Fall 2014

Computing with Spintronics: Circuits and architectures

Rangharajan Venkatesan
Purdue University

Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Rangharajan Venkatesan

Entitled
Computing with Spintronics: Circuits and Architectures

For the degree of Doctor of Philosophy

Is approved by the final examining committee:

ANAND RAGHUNATHAN
Chair

ARIJIT RAYCHOWDHURY

BYUNGHOO JUNG

KAUSHIK ROY

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): ANAND RAGHUNATHAN

Approved by: M. R. Melloch 10-01-2014
Head of the Graduate Program Date

COMPUTING WITH SPINTRONICS: CIRCUITS AND ARCHITECTURES

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Rangharajan Venkatesan

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

December 2014

Purdue University

West Lafayette, Indiana

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my advisor, Prof. Anand Raghunathan for his outstanding support throughout my Ph.D. His excellent guidance enabled me to explore several different research topics of my Ph.D. with relative ease. His constant motivation and encouragement played a significant role in overcoming various hurdles and was a great inspiration towards achieving different milestones during my Ph.D. I would also like to thank him for creating a diverse learning atmosphere in the lab, which enabled me to acquire knowledge across a wide spectrum in both technical and non-technical areas.

I would like to thank Prof. Kaushik Roy for his support and mentorship. His advice and motivation was a great encouragement for me towards pursuing exciting ideas during my Ph.D. I would like to thank Prof. Byunghoo Jung and Prof. Arijit Raychowdhury for serving in my doctoral committee and for providing insightful thoughts and feedback during my Ph.D.

My sincere thanks to the alumni and current members of Integrated Systems Laboratory (ISL) and Nanoelectronics Research Laboratory (NRL) for their collaboration and friendship during my Ph.D. Special thanks to my parents, Venkatesan and Jailakshmi, and my sister Sundharavalli for their support, love and affection throughout my entire life.

TABLE OF CONTENTS

Page

Page

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABSTRACT

Venkatesan, Rangharajan Ph.D., Purdue University, December 2014. Computing with Spintronics: Circuits and Architectures. Major Professor: Prof. Anand Raghunathan.

With the scaling of semiconductor technology approaching its fundamental limits, several potential replacements are being actively explored to the mainstays of Silicon and CMOS. Among them, spintronics, which uses electron 'spin' as the state variable to represent and process information, has attracted significant interest in recent years.

Spintronic devices are considered promising due to their non-volatility, near-zero leakage, and high integration density, all of which compare favorably to CMOS. However, they are not superior in all respects. Spintronic memories such as Spin-Transfer Torque Magnetic RAM (STT-MRAM) require high write energy and latency. Spintronic logic proposals such as All-Spin Logic (ASL) are limited by large current requirement for high speed operation, high short circuit power and low spin-diffusion length of spin interconnects. Therefore, spintronic devices are neither universal nor drop-in replacements. This creates the need for new designs at the circuit and architecture levels that exploit the strengths of spintronic devices and mask their weaknesses.

This thesis makes the following contributions towards the design of computing platforms with spintronic devices.

- It explores the use of spintronic memories in the design of a domain-specific processor for an emerging class of data-intensive applications, namely recognition, mining and synthesis (RMS). Two different spintronic memory technologies—Domain Wall Memory (DWM) and STT-MRAM—are utilized to realize the different levels in the memory hierarchy of the domain-specific processor, based

on their respective access characteristics. Architectural tradeoffs created by the use of spintronic memories are analyzed. The proposed design achieves 1.5X-4X improvements in energy-delay product compared to a CMOS baseline.

- It describes the first attempt to use DWM in the cache hierarchy of general-purpose processors. DWM promises unparalleled density by packing several bits of data into each bit-cell. TapeCache, the proposed DWM-based cache architecture, utilizes suitable circuit and architectural optimizations to address two key challenges (i) the high energy and latency requirement of write operations and (ii) the need for shift operations to access the data stored in each DWM bit-cell. At the circuit level, DWM bit-cells that are tailored to the distinct design requirements of different levels in the cache hierarchy are proposed. At the architecture level, TapeCache proposes suitable cache organization and management policies to alleviate the performance impact of shift operations required to access data stored in DWM bit-cells. TapeCache achieves more than 7X improvements in both cache area and energy with virtually identical performance compared to an SRAM-based cache hierarchy.

- It investigates the design of the on-chip memory hierarchy of general-purpose graphics processing units (GPGPUs)—massively parallel processors that are optimized for data-intensive high-throughput workloads—using DWM. STAG, a high density, energy-efficient Spintronic-Tape Architecture for GPGPU cache hierarchies is described. STAG utilizes different DWM bit-cells to realize different memory arrays in the GPGPU cache hierarchy. To address the challenge of high access latencies due to shifts, STAG predicts upcoming cache accesses by leveraging unique characteristics of GPGPU architectures and workloads, and prefetches data that are both likely to be accessed and require large numbers of shift operations. STAG achieves 3.3X energy reduction and 12.1% performance improvement over CMOS SRAM under iso-area conditions.

- While the potential of spintronic devices for memories is widely recognized, their utility in realizing logic is much less clear. The thesis presents SPINTASTIC, a new paradigm that utilizes Stochastic Computing (SC) to realize spintronic logic. In SC, data is encoded in the form of pseudo-random bitstreams, such that the probability of a '1' in a bitstream corresponds to the numerical value that it represents. SC can enable compact, low-complexity logic implementations of various arithmetic functions. SPINTASTIC establishes the synergy between stochastic computing and spin-based logic by demonstrating that they mutually alleviate each other's limitations. On the one hand, various building blocks of SC, which incur significant overheads in CMOS implementations, can be efficiently realized by exploiting the physical characteristics of spin devices. On the other hand, the reduced logic complexity and low logic depth of SC circuits alleviates the shortcomings of spintronic logic. Based on this insight, the design of spin-based stochastic arithmetic circuits, bitstream generators, bitstream permuters and stochastic-to-binary converter circuits are presented. SPINTASTIC achieves 7.1X energy reduction over CMOS implementations for a wide range of benchmarks from the image processing, signal processing, and RMS application domains.

- In order to evaluate the proposed spintronic designs, the thesis describes various device-to-architecture modeling frameworks. Starting with devices models that are calibrated to measurements, the characteristics of spintronic devices are successively abstracted into circuit-level and architectural models, which are incorporated into suitable simulation frameworks.

The results presented in this thesis suggest that spintronics can add significant value to the design of future computing platforms. Spintronic devices are well-suited for designing memories, but suitable circuits and architectures (as proposed in this thesis) can significantly enhance their benefits. On the other hand, although spintronic logic faces fundamental challenges as a drop-in replacement for CMOS, efficient im-

plementations may be possible for some application domains whose computational characteristics match those of spintronic devices.

1. INTRODUCTION

In the history of the semiconductor industry, an endeavor to deliver compact, high performance systems at low power has been the major driving force. Figure 1.1 shows a timeline of some of the important advances in the device technology during the evolution of the semiconductor industry. Historically, technology scaling, which involved shrinking of device dimensions and voltage scaling simultaneously, was the major source of the key benefits in terms of density, power and performance. However, the industry has seen multiple tipping points when a fundamental change in the device technology was required to achieve sustained benefits. These include the transition from vacuum tubes to Bipolar Junction Transistors (BJTs) in the late 1940s and the adoption of Complementary Metal Oxide Semiconductor (CMOS) technology in the 1980s. As CMOS technology scales deep into the nanometer regime, further scaling is facing a number of challenges due to increased variations, higher leakage power, *etc.* and many researchers believe that the integrated circuit industry is nearing another such tipping point. This has led to an intense quest for identifying the "next switch", and several new devices (*e.g.* III-V tunnel FET (TFET) [1], nano-electro-mechanical switches (NEMS) [2], carbon nanotube (CNT) [3], graphene nanowire [4], spintronic devices [5–7], *etc.*) have been proposed as potential CMOS replacements.

A promising direction that has emerged from this quest is *spintronics*, which uses spin rather than charge as the state variable to represent and process information. Spintronic devices are considered highly promising due to their non-volatility (they can retain data even when the power supply is switched off leading to very low leakage power), and high integration density resulting in smaller area footprint compared to CMOS. Spin-based devices such as Spin-Transfer Torque Magnetic RAM (STT-MRAM) [10–13], domain wall memory (DWM) [14–17] *etc.* are widely considered as promising candidates for future on-chip and off-chip memory. Several industrial

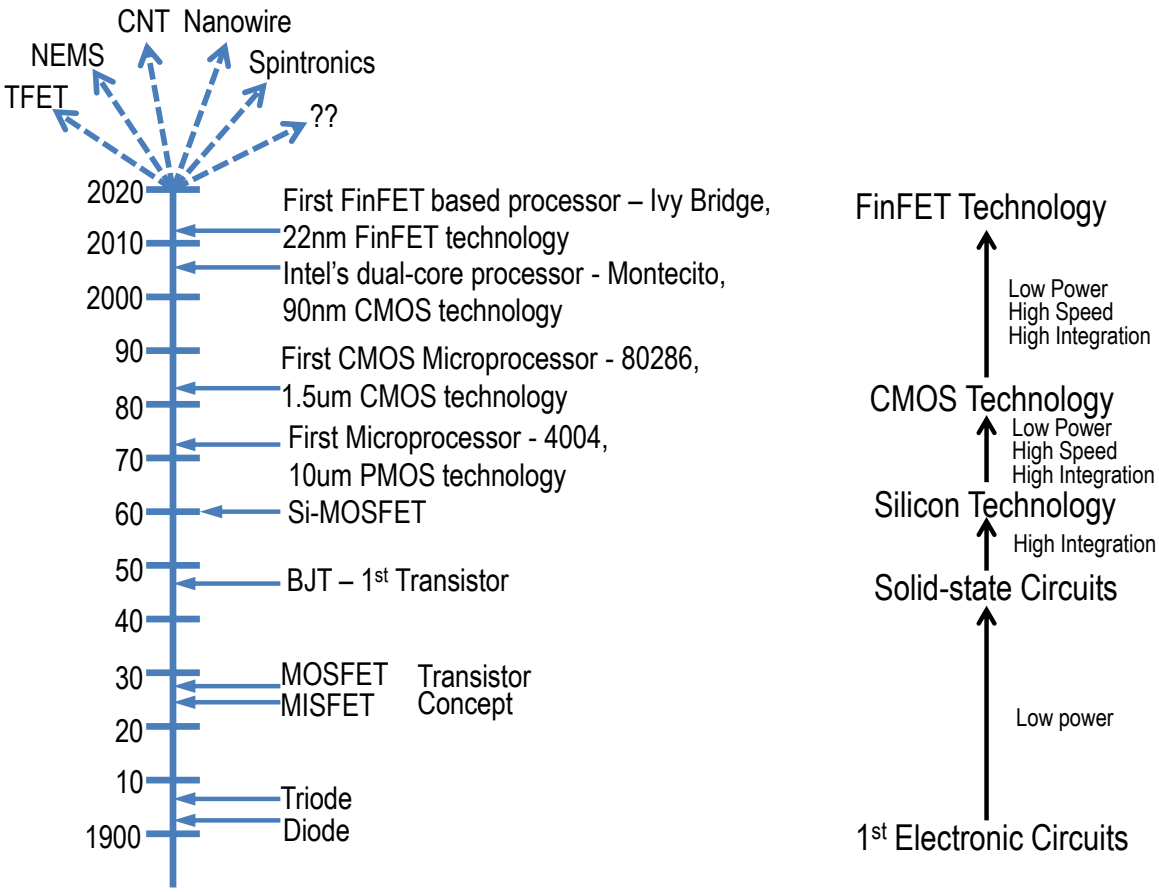


Fig. 1.1.: Evolution of semiconductor industry [8, 9]

prototypes and early commercial products have underscored the promise of spin-based memories [18–22]. Recent research efforts have shown the feasibility of realizing logic functionality (both Boolean and non-Boolean) using spin-based devices [23–29]. While spintronic devices are highly promising, a comprehensive circuit-to-architecture study is required to explore the design of future spin-based computing platforms. This thesis attempts to perform such a study.

In the following sections, we first present an overview of current CMOS scaling challenges. We then describe some of the promising spintronic device technologies and motivate the need for new circuits and architectures from both the device as well as application perspective. Finally, we conclude this chapter with a brief description of the contributions of this thesis and an overview of the remaining chapters.

1.1 CMOS scaling challenges

Scaling of CMOS technology, in accordance with Moore's law, has provided with recurring benefits in terms of increased processing capabilities, higher performance and lower energy consumption. However, continued scaling into the nanometer regime has thrown up a number of challenges such as increased process variations, higher leakage power, decrease in reliability of devices, *etc.* Let us examine these issues in more detail.

1.1.1 Process variations

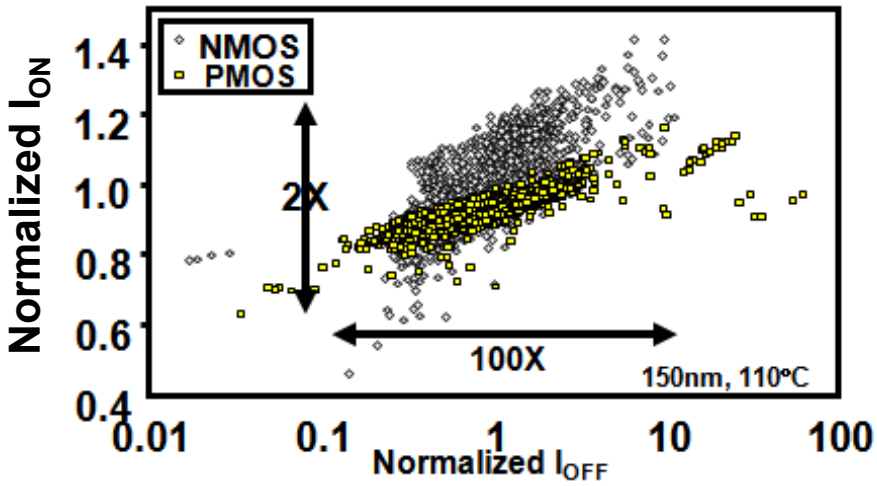


Fig. 1.2.: Variation in I_{ON} and I_{OFF} of transistors at the 65nm node [30]

The inability of the lithographic process to precisely control the process parameters has resulted in process variations becoming a major challenge to the continued scaling of CMOS technology [31, 32]. Variations in parameters like channel length, thickness of gate oxide, dopant concentration, *etc.* cause corresponding changes in the transistor's electrical characteristics like threshold voltage, eventually resulting in variations in the transistor switching characteristics. Figure 1.2 illustrates the impact of process variations on the ON and OFF currents of the transistors at the 65nm

node. We can see that the leakage current varies by about two orders of magnitude, while ON current varies by a factor of two.

Increased variations force the designers to introduce design guard bands in order to ensure correct functionality even under worst case conditions, thereby resulting in over-design of the systems. As device dimensions shrink further with scaling of technology, variations become more significant and require larger and larger guard bands. This could eventually negate all the benefits of scaling, thereby bringing about the end of CMOS technology scaling.

1.1.2 Power and Temperature

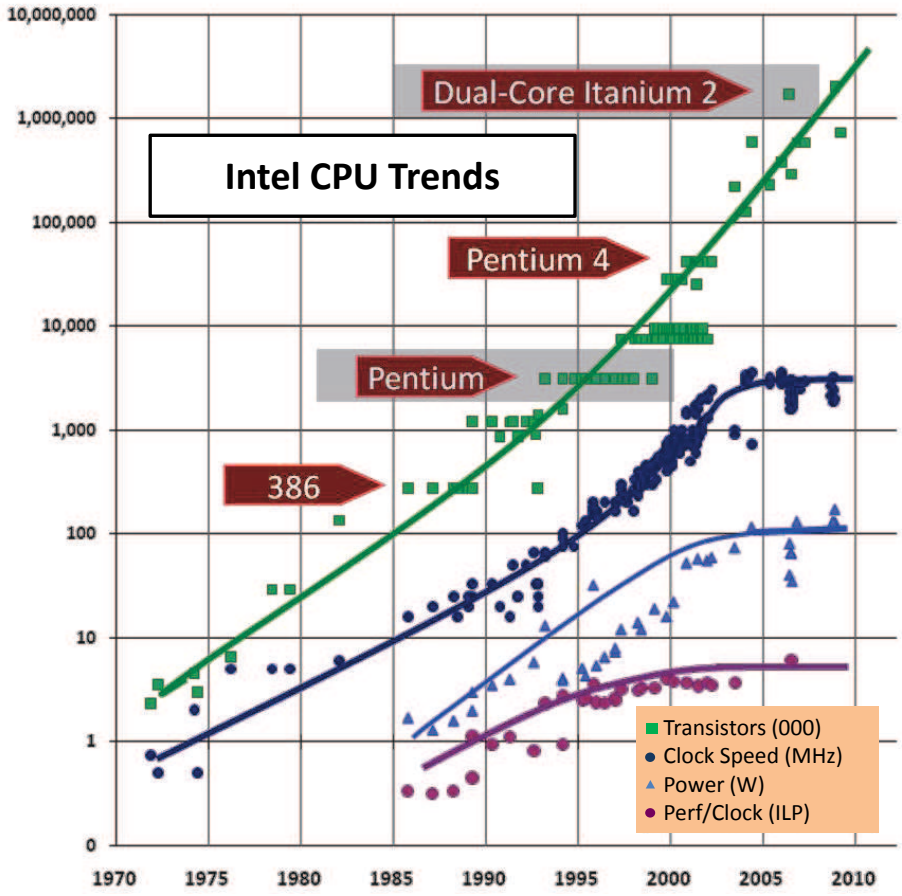


Fig. 1.3.: Power and performance trend in Intel microprocessors [33]

Scaling of technology resulted in exponential increase in both the number of transistors integrated per chip as well as the frequency of their operation. Figure 1.3 shows the trends in transistor count and frequency for Intel microprocessors. These have been the major contributors to the improvement in performance of the system till early 2000s. However, the increase in the transistor count and frequency also resulted in an increase in the power consumption of chips, as shown in Figure 1.3. With the total power consumption reaching the power budget of 100W, the scaling trend had already changed causing tectonic shifts (such as the switch to parallelism rather than clock frequency as the primary driver of performance in microprocessors) [34].

Another major challenge to the scaling of CMOS technology is the increase in power density and the corresponding increase in the on-chip temperature. Traditional scaling of CMOS technology involved scaling of threshold voltage and the supply voltage along with transistor dimensions, resulting in power density remaining constant. However, with the increase in short channel effects and the corresponding increase in the leakage current, the threshold voltage cannot be scaled at the same rate as that of the transistor dimensions. This in turn has slowed down the rate of supply voltage scaling in order to maintain performance, resulting in an increase in the power density of the chip. Increase in power density causes the on-chip temperature to rise leading to thermal concerns.

1.1.3 Reliability

The reliability concerns of CMOS technology arise from a number of factors such as hot carrier injection (HCI), negative bias temperature instability (NBTI), time dependent dielectric breakdown (TDDB), electromigration, *etc.* HCI occurs when an high energy electron gets trapped in the gate oxide. NBTI is caused due to generation of positive oxide charge and interface traps in PMOS under negative bias conditions [35]. HCI and NBTI causes the threshold voltage of transistors to increase, resulting in performance degradation. TDDB is a phenomenon that is caused

by the formation of conducting paths through the gate oxide due to electron tunneling current. Unlike HCI or NBTI, which result in gradual degradation, TDDB is a catastrophic failure causing the circuit to malfunction. Electromigration is the transport of material caused by the gradual movement of the ions in a conductor, resulting in increases in the resistances of the interconnects.

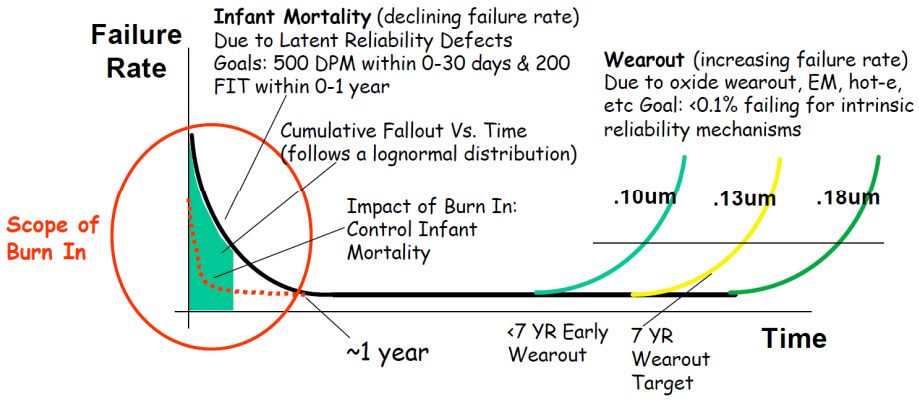


Fig. 1.4.: Lifetime degradation with CMOS scaling [36]

With scaling of CMOS technology, the effect of NBTI, HCI, TDDB and electromigration increases, resulting in considerable reliability degradation. This results in reduction in the lifetime with each technology generation, as shown in Figure 1.4.

As a result of the above mentioned challenges, the scaling of CMOS technology is likely to come to an end within the next decade. This has led to the search for alternate device technologies that can complement or potentially replace the existing CMOS technology. Spintronics is widely considered to be highly promising and we present a brief description of it next.

1.2 Spintronics

Spintronics is an emerging technology in which the processing, storage, and communication of data is performed by means of electron-spin along with or instead of electron-charge. The basic unit in a spintronic device is a magnet that can have two different magnetization directions representing 0 and 1. A number of different

spintronic devices have been proposed for realizing both memory and logic. In the following sections, we describe some of the promising spintronic memory and logic technologies.

1.2.1 Spintronic Memory

With the increase in contribution of on-chip memory to the total chip area and power consumption, there has been a surge in the number of research efforts to identify suitable replacements for CMOS-based memories (SRAM and DRAM). Several new memory technologies— Ferroelectric RAM (FeRAM), Phase Change RAM (PCRAM), STT-MRAM, DWM, *etc.* have been proposed as potential replacements. Among them, spintronic memory technologies (STT-MRAM and DWM) are considered as potential candidates for future on-chip memory designs due to their high density, non-volatility leading to low leakage power and read access latencies that are comparable to SRAM/DRAM. In the following sections, we present a brief description of the two promising spintronic memory technologies – STT-MRAM and DWM – and analyze their benefits and challenges.

Spin-Transfer Torque Magnetic RAM

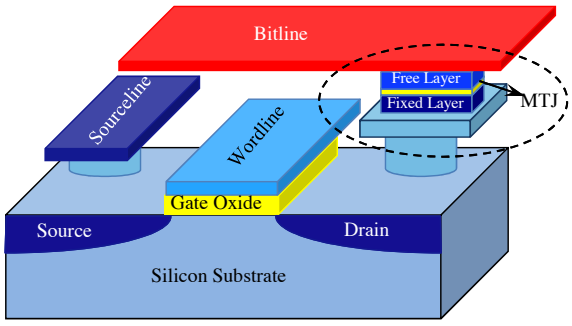


Fig. 1.5.: Spin-Transfer Torque Magnetic RAM

Spin-Transfer Torque Magnetic RAM or STT-MRAM is a spintronic memory technology that stores data using a magnetic tunneling junction (MTJ). Figure 1.5 shows the structure of an STT-MRAM cell consisting of an MTJ and an access transistor. An MTJ consists of two ferromagnetic layers (free layer and fixed layer) separated by a thin dielectric tunneling barrier. The magnetization of the fixed layer is constant while that of the free layer can be varied using a spin-polarized current. Depending on the relative magnetic orientation between the free layer and the fixed layer, an MTJ offers different resistances that are used to represent '0' (low resistance) and '1' (high resistance), respectively in an STT-MRAM cell. STT-MRAM has the desirable characteristics of high density, low leakage power as well as very high endurance. However, the high write energy and write latency of STT-MRAM are major bottlenecks.

Domain Wall Memory

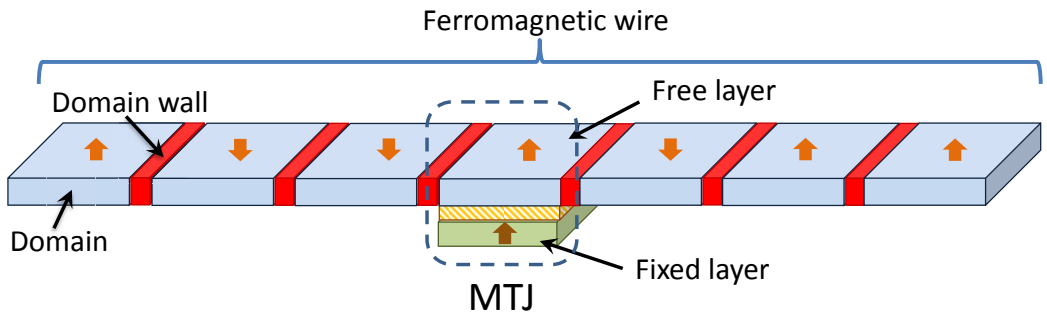


Fig. 1.6.: Domain Wall Memory

Domain Wall Memory (DWM) is another spin-based memory technology in which bits of data are densely packed in the domains of a ferromagnetic wire. A simple DWM device consisting of a ferromagnetic wire along with a read/write port (usually an MTJ) is shown in Figure 1.6. A key feature of DWM is that bits stored in the ferromagnetic wire can be shifted by applying a current pulse. This enables DWM to share the read/write port across multiple bits in the ferromagnetic wire and achieve much higher density compared to SRAM, DRAM, and even other emerging memory

technologies. However, this also introduces the need for performing shift operations for accessing the bits, which increases the access latency of DWM.

1.2.2 Spintronic Logic

While the utility of spintronic devices as a viable memory technology has been adequately established, a few recent efforts viz. Nano-Magnetic Logic (NML) [23], All-Spin Logic (ASL) [24], have explored its effectiveness in realizing logic functionality. These logic styles possesses several key attributes: (i) non-volatility resulting in near-zero leakage power, (ii) compact logic implementations resulting in small area footprint. In this section, we present a brief description of these logic styles and analyze their benefits and limitations.

Nano-Magnetic Logic

Nano-Magnetic Logic (NML) is a spintronic logic technology that uses dipolar coupling across neighboring magnets to realize logic functions. An example of a 3-input NML gate is shown in Figure 1.7. Initially, all the magnets are reset using an external magnetic field. Then, during evaluation, the external magnetic field is removed and the input magnets are switched to the required magnetic orientation. This causes switching of the neighboring magnets through dipole interaction, eventually resulting in the output magnet to be switched to the desired orientation. In this way, the output of the NML gate can be evaluated without any current.

While NML offers benefits in terms of non-volatility and density, it suffers from several major limitations. NML works on the principle of near-neighbor communication, which severely limits its applicability to a wide range of applications and architectures. They require external magnetic field whose generation causes significant energy overheads. It also affects technology scaling as it is challenging to concentrate the external field on to a magnet. Also, the required critical magnetic field increases with scaling.

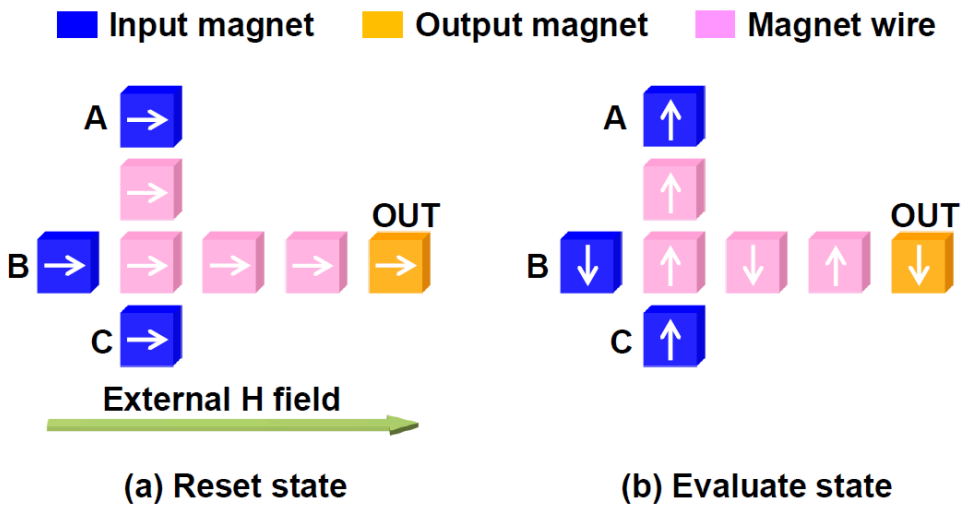


Fig. 1.7.: Nano-Magnetic Logic

All-Spin Logic

All-spin logic is a recently proposed logic design paradigm for spintronic devices that uses spin-polarized current and spin-torque switching mechanism to switch a magnet between its spin states. An example of a ASL inverter consisting of two spin magnets (input magnet and output magnet) connected through a spin channel is shown in Figure 1.8. Initially, assume that both the input and output magnets are oriented in the same direction. When a current is injected into the input magnet as shown in Figure 1.8, electrons whose spin orientation is opposite to that of the input magnet gets accumulated in the channel. This leads to a spin potential difference in the channel, causing a spin current to flow through it. This current exerts a spin-torque on the output magnets and switches it opposite to the magnetic orientation of the input magnet.

ASL overcomes some of the major limitations of NML. The communication in ASL is not limited to near-neighbor communication as it uses spin currents to propagate signals through a spin channel. In addition, the use of spin-torque switching mechanism makes ASL highly scalable. However, ASL also faces certain drawbacks.

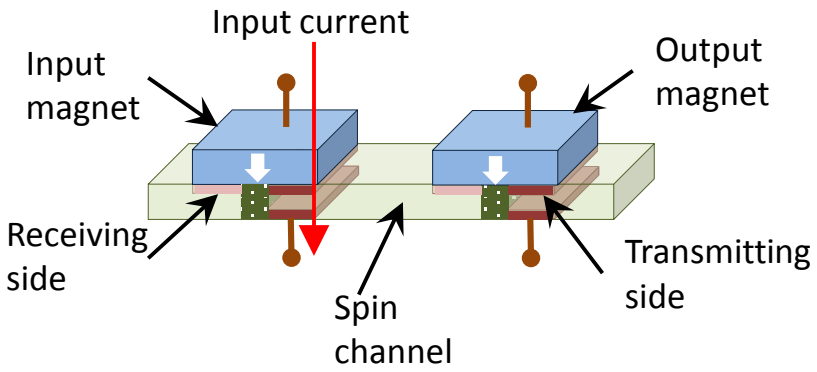


Fig. 1.8.: All-Spin Logic

Injecting current through the metallic spin-magnets results in very high short circuit power. Also, cascading of multiple logic gates significantly increases the delay of the circuit, resulting in high energy consumption. Further, the low spin-diffusion length of spin channels is also a major limitation.

1.2.3 Motivation for new circuits and architectures

From the above discussions, we find that spintronic devices are fundamentally different, and their characteristics vary significantly, from their CMOS counterparts.

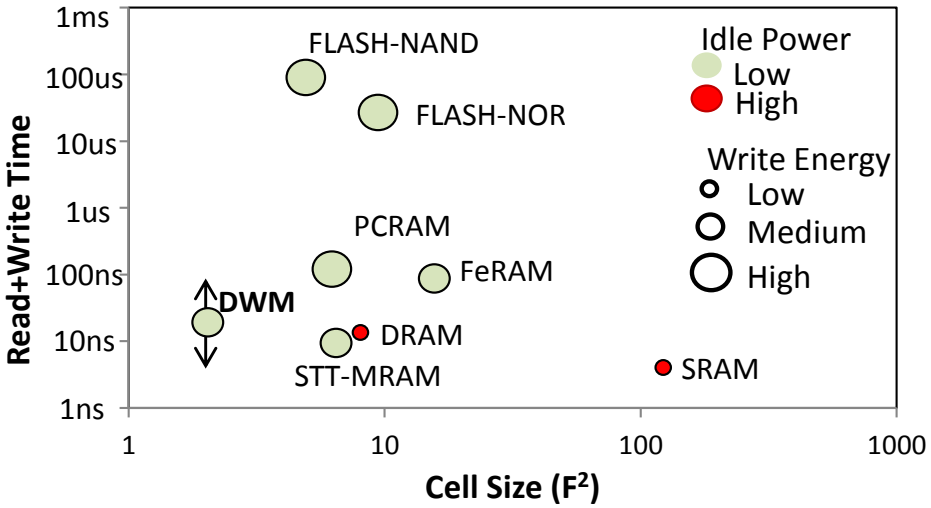


Fig. 1.9.: Comparison of different memory technologies [14]

In the context of spin memories, Figure 1.9 presents a comparison of different key metrics with other memory technologies. As shown in the figure, the spintronic memories (STT-MRAM and DWM) possess a number of desirable characteristics that make them promising candidates for future memory designs. They offer higher density and have lower leakage power due to their non-volatile nature compared to SRAM and DRAM. In particular, DWM offers the highest density and it outperforms even other emerging memory technologies like PCRAM and STT-MRAM. Further, the access latency of these spin-based memories is much lower than other non-volatile memories like PCRAM and Flash, making them promising on-chip memory candidates. However, spintronic memories also have certain limitations. The write energy and latency of STT-MRAM and DWM are much higher than SRAM/DRAM. In the case of domain wall memory, very high density is achieved by sharing of access transistors across multiple bits in a single cell. As a result, accessing a bit from DWM bit-cell requires shift operations leading to variable (potentially high) access latencies.

When we consider logic design using spintronic devices, proposals like ASL offer benefits over CMOS due to their non-volatility and ability to realize compact logic implementations with lower device count. This leads to lower leakage power and smaller area footprint than CMOS. However, ASL has high short circuit power, which leads to significantly higher energy consumption compared to CMOS. Further, the small spin-diffusion length of spin channels leads to highly inefficient interconnect designs. Figure 1.10 shows a comparison of the area and energy consumption of ASL

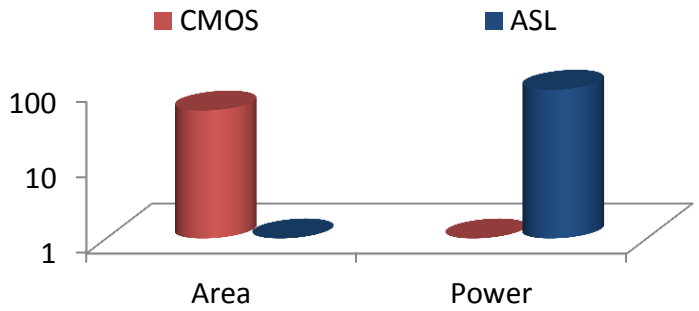


Fig. 1.10.: Comparison of area and power consumption of ASL with CMOS for Leon-Sparc processor

with CMOS for LeonSparc processor operating at a frequency of 1GHz. As shown in the figure, ASL can enable significant area benefits, but has very high energy consumption due to high short circuit power and inefficient interconnects.

This necessitates the need to design suitable circuits and architectures that would exploit the strengths of these emerging devices and mask their weaknesses.

From an application perspective, the continuous growth in the complexity of algorithms and the amount of data they process have fueled an ever-increasing need for larger processing power. This in turn has resulted in an increased demand for both processing cores as well as on-chip memories as illustrated in Figures 1.11 and 1.12, respectively. In particular, the on-chip memories have witnessed an exponential

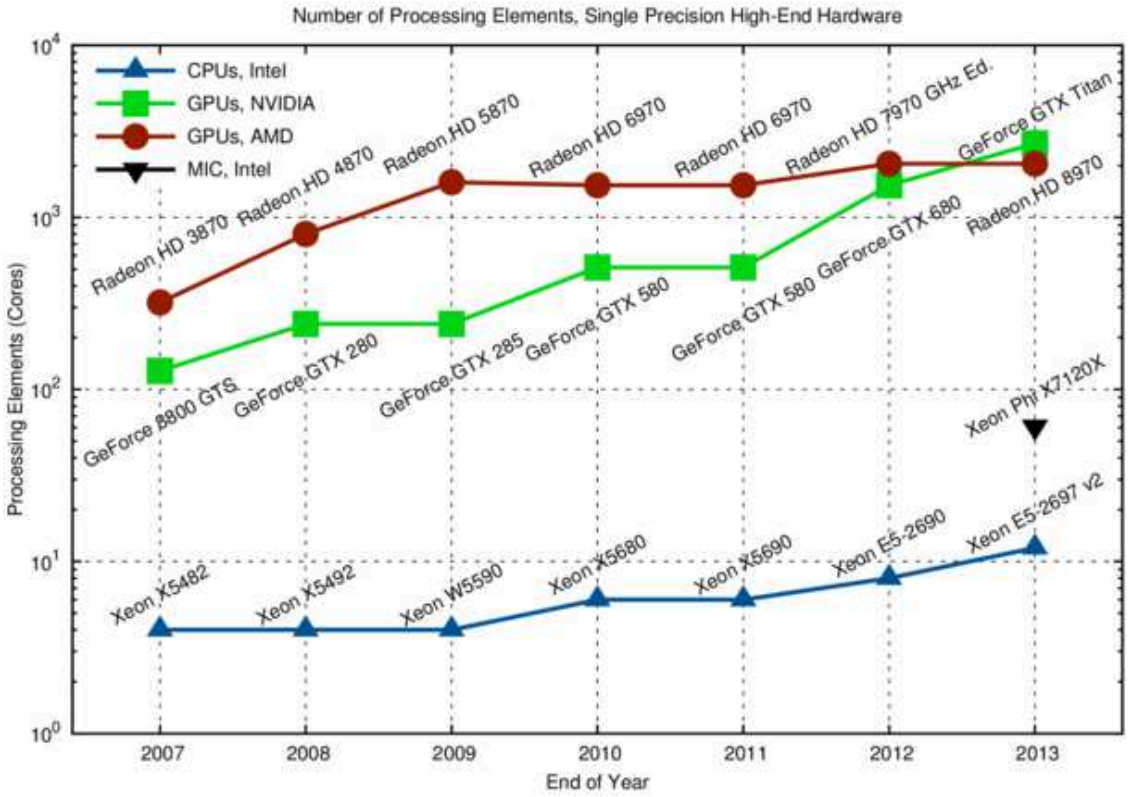
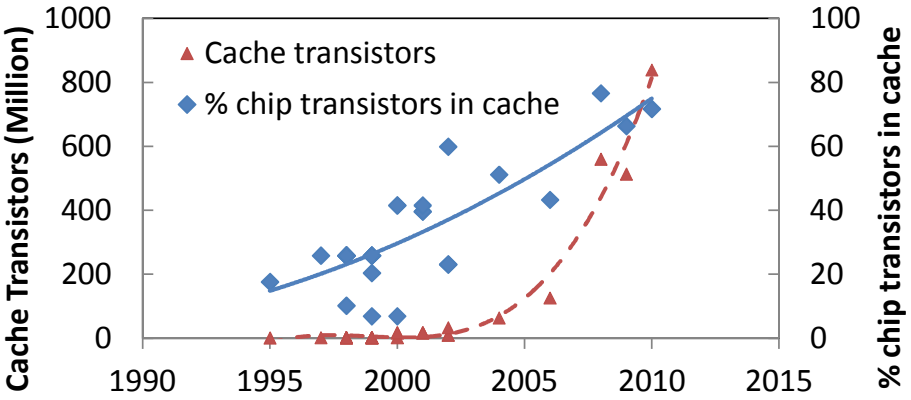
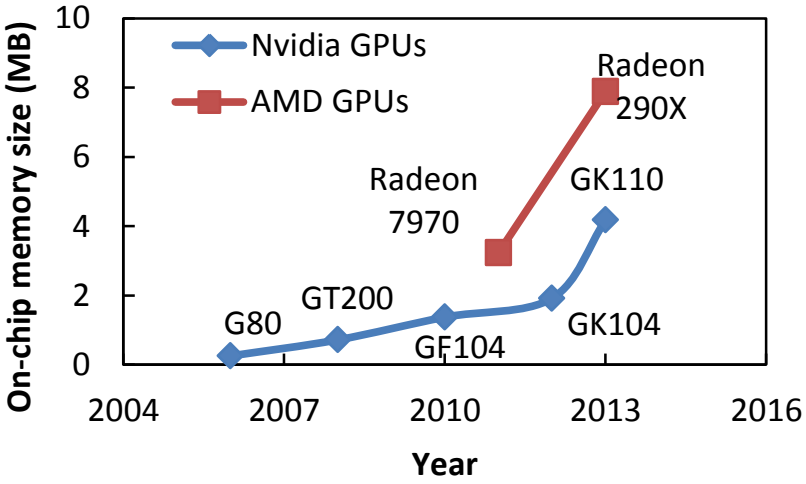


Fig. 1.11.: Growth in number of processing elements or cores in CPUs and GPUs [37]

growth, making them one of the major contributors to chip area, transistor count, as well as power consumption. The move to multi-cores, and the integration of proces-



(a) On-chip memory trend in CPUs



(b) On-chip memory trend in GPUs

Fig. 1.12.: Growth in on-chip memory of CPUs and GPUs [30, 38, 39]

sors with GPUs in the modern products (like Intel Sandy bridge, AMD Fusion) has further enhanced the need for larger on-chip memories.

Recent years have also seen the emergence of a new class of applications – Recognition, Mining and Synthesis. These applications are highly data-intensive and are expected to drive the demand for specialized processing elements and greater amounts of on-chip memory [40]. As a result, spintronic devices that offer high density at low power are of great interest. However, in order to fully exploit the potential of these devices, we need to design suitable circuits and architectures that match the distinct characteristics of these devices with the demands of the applications.

1.3 Thesis overview

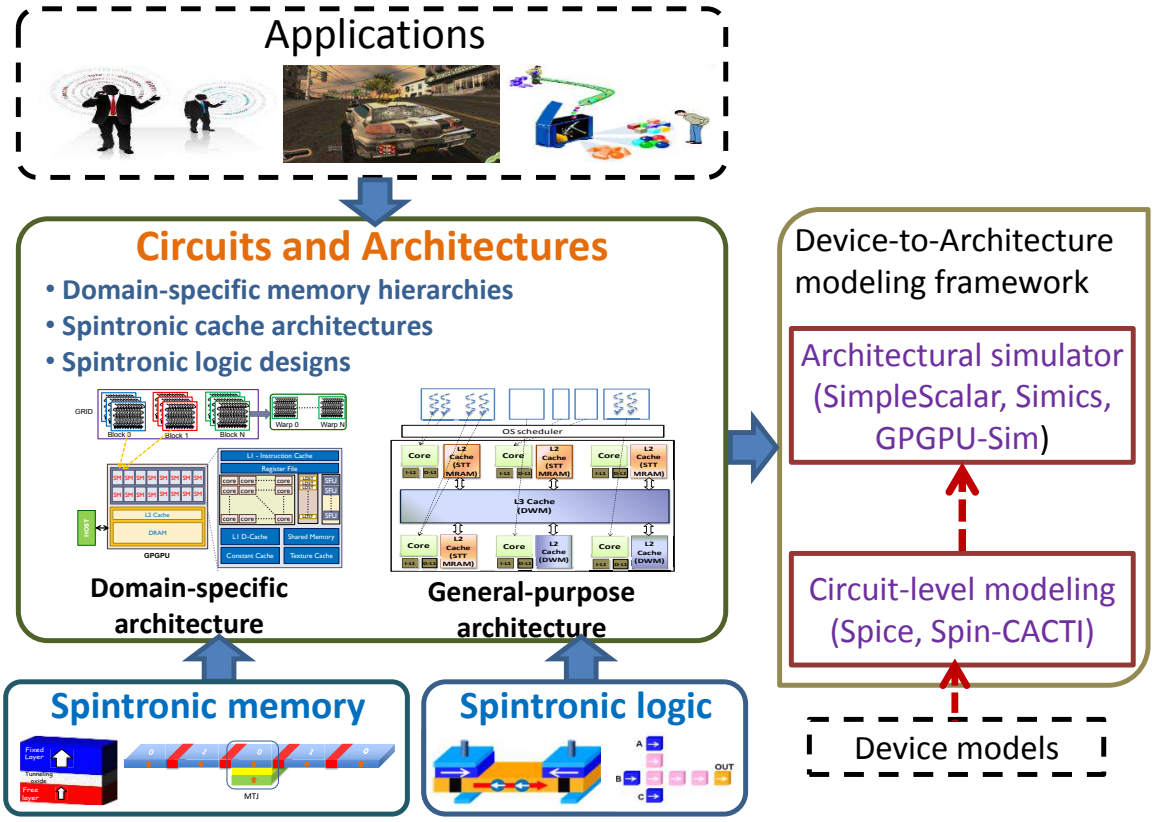


Fig. 1.13.: Thesis overview

In this thesis, we focus on the design of suitable circuits and architectures for next generation computing platforms using spintronic device technologies. We explore the design of spin-based memory architectures as well as logic for computing platforms across a spectrum from single core micro-processors to many-core graphics processing units (GPUs) and domain-specific accelerators as shown in Figure 1.13.

We perform the first exploration of the design of domain wall memory based cache in general-purpose architectures. We propose a novel all-spin cache design in which we use spin-based memories to design all the levels in the cache hierarchy including the L1 cache, where spin-based memories have hitherto not been used due to their high write latency/energy. The proposed design incorporates various circuit

and architectural optimizations to address the challenges associated with the use of spintronic memories at different levels in the cache hierarchy.

We explore the design of memory hierarchy of a domain-specific many-core accelerator using spin-based memories. We leverage the flexibility offered by domain-specific architectures to tune the architectural parameters to suit the demands of applications as well as device characteristics. We demonstrate that synergistically performing suitable circuit and architectural optimizations along with the use of appropriate spintronic memory technologies can lead to significant benefits.

We explore a new computing paradigm for designing logic with spintronic devices. We propose the use of Stochastic Computing (SC) to realize spintronic logic designs. In SC, the data is encoded as pseudo-random bitstreams such that the probability of a '1' in the bitstream represents its magnitude. SC can enable compact implementations of various arithmetic functions such as adders, multipliers, *etc.* However, it requires additional hardware for stochastic bitstream generators, bitstream permuters and stochastic-to-binary converters, which incur significant overheads in CMOS. We demonstrate that there exists a synergy between SC and spintronic logic, *i.e.,* they alleviate each other's limitations.

In order to model and evaluate the proposed designs, we have developed a detailed device-to-architecture modeling framework, as illustrated in Figure 1.13. The framework takes experimentally validated device models of spintronic devices as input and abstracts them into circuit and architectural models for evaluating and exploring different spin-based designs.

1.4 Thesis organization

The rest of this thesis is organized as follows. In Chapter 2, we present a brief survey of prior research efforts that have focused on various design issues related to emerging technologies. In Chapter 3, we provide background information related to

the structure and operation of spin-based memories (STT-MRAM and DWM) and spintronic logic at the device and circuit levels.

In Chapter 4, we describe the design of memory hierarchy of the domain-specific many-core processor architecture for RMS applications using spin-based memories. The proposed design consists of a two dimensional array of processing elements along with a two level on-chip memory hierarchy. The first level is formed by an array of FIFO memory units that are responsible for providing fast streaming access to data. The second level in the memory hierarchy is a random access memory that stores a sizable part of the data set being processed. Based on these design requirements and the memory device characteristics, we suggest the use of DWM and STT-MRAM to realize the first and second levels, respectively. We evaluate the proposed design for three representative recognition and mining algorithms, namely Support Vector Machines (SVM), k-means clustering and Generalized Learning Vector Quantization (GLVQ). Our analysis shows that the proposed domain-specific architecture that is tuned to match the device characteristics with application requirements can result in 1.5X-4X improvement in energy-delay product compared to CMOS baseline.

In Chapter 5, we propose Tapestri, in which we address the challenge of high write energy and write latency associated with the MTJ-based write mechanism in spin-based memories. Our proposal is based on the observation that domain wall shifts offers an efficient mechanism to perform write operations. We exploit this fact to design different bit-cell designs, 1bitDWM and multibitDWM, that are optimized for latency and area, respectively. We explore various circuit-level optimizations for the proposed bit-cells. We show that 1bitDWM can achieve all the benefits offered by STT-MRAM, while matching SRAM in its write efficiency. MultibitDWM, on the other hand, achieves much higher density than SRAM, STT-MRAM and 1bitDWM at the cost of variable access latencies.

In Chapter 6, we present TapeCache, in which we make the first attempt to design the cache hierarchy of general-purpose processor using DWM. In this work, we address one of the major design challenges with DWM – the performance penalty

due to sequential accesses to data stored in DWM. We propose a circuit-architecture co-design technique consisting of (i) multi-port read-skewed multibitDWM bit-cell design at the circuit level and (ii) hybrid cache organization and suitable management policies at the architecture level to maximally harness the performance potential of DWM. Our multi-port read-skewed multibitDWM bit-cell design exploits the read-write asymmetry in cache accesses to reduce the access latency of performance critical read operations. Our cache management policies exploits the spatial locality property to reduce the impact of sequential accesses on the overall performance of the system. TapeCache achieves 7.5X improvement in energy and 7.8X reduction in area at virtually identical perform compared to an iso-capacity CMOS SRAM baseline.

In Chapter 7, we propose STAG, a Spintronic-Tape Architecture for GPGPU cache hierarchy. STAG employs different DWM bit-cells to realize different memory arrays in the GPGPU cache hierarchy based on their design requirements. To address the performance penalty associated with shift operations required to access data from multibitDWM bit-cell, STAG utilizes suitable architectural optimizations that predicts the cache access patterns based on the unique characteristics of GPGPU architecture and workloads, and prefetches data that are both likely to be accessed and require large number of shifts. STAG achieves 3.3X energy reduction and 12.1% performance improvement over SRAM-based cache under iso-area conditions.

In Chapter 8, we focus on the design of spin-based logic. We present SPINTASTIC, in which we propose stochastic computing (SC) as a new direction to realize logic using spin-based devices. We establish the synergy between SC and spintronic logic by demonstrating that their characteristics mutually benefit each other. We show that the physical characteristics of spin devices enable efficient realization of different key components in stochastic logic circuits, while the low logic complexity and logic depth of SC can in-turn mitigate some of the drawbacks of spintronic logic. Our experiments shows that SPINTASTIC achieves 7.1X energy reduction over CMOS implementations.

In Chapter 9, we describe the modeling framework that is used to evaluate the proposed designs. We present various self-consistent device models that have been

validated with experimental data that are used to evaluate different spintronic devices. We also describe Spin-CACTI – a CACTI based cache simulator that computes various performance metrics of spin-based caches.

Finally, Chapter 10 concludes this thesis. In this chapter, we revisit the key benefits offered by the use of spintronic devices to design computing platforms and summarize the key findings.

2. RELATED WORK

CMOS technology is reaching its fundamental scaling limits and several new devices have been proposed as potential replacements. These include emerging memory technologies like phase change memory, spin-based memories (STT-MRAM, DWM), memristors, etc. and logic switches such as Tunnel FET (TFET), Bilayer pseudospin FET (BiSFET), Carbon Nanotubes (CNT), Spin Wave Device (SWD), *etc.* In the past decade, there has been increasing interest to explore and address some of the key issues related to designing with these devices at various levels of design abstraction. In this chapter, we describe some of the significant efforts in this direction.

2.1 Emerging memory technologies

With the increase in application complexity and data set size, the contribution of memories (on-chip and off-chip) to the total energy consumption of the system is on the rise. This has lead to a number of research efforts that have investigated the use of various emerging memory technologies as potential CMOS memory replacements. Some of these efforts have shown STT-MRAM and PCRAM as promising candidates for implementing cache and main memory respectively [11–13, 41–46]. While STT-MRAM and PCRAM have desirable properties like high-density and non-volatility, they also have drawbacks such as high write energy and high write latency that need to be addressed. In addition, the limited endurance of PCRAM is a major concern. These issues were studied in detail and a number of optimizations have been proposed at the device, circuit and architecture levels. Other technologies like domain wall memory, memristor, *etc.* have also attracted significant interest [14, 15, 47–51].

2.1.1 Device and circuit optimizations

At the device level, researchers have optimized the write operation in STT-MRAM by designing different kinds of MTJ structures such as dual-pillar MTJ, tilted MTJ, dual-barrier MTJ, *etc.* [52–54]. Many of these device proposals decouple read/write paths, thereby relaxing the read *vs.* write design conflicts that are commonly present in memory design. Another approach at the device level involves exploring newer switching mechanisms like thermally-assisted-STT switching, resonant switching, *etc.* [55–59] to optimize write operations. At the circuit level, proposals to use 2T-1R structures with dual source line, adaptive bitline biasing, early write termination are aimed at reducing the total write energy consumption [60–63]. In addition, research efforts at the circuit level have also focused on analyzing the impact of process variations [60, 64], improving the read latency by designing efficient sensing schemes [65] and enhancing the density of the cell through multi-level STT-MRAM designs [66]. In [67], the authors studied the design of energy-efficient and robust STT-MRAM arrays and showed the importance of considering the array level tradeoffs on the stability and energy efficiency of STT-MRAM. In the case of PCRAM, structures like μ-trench [68, 69], wall [70, 71], cross spacer [72], edge [73], *etc.* have been proposed to address the high write current. The endurance problem with PCRAM was primarily addressed through doping of the phase change material [74]. In [75], the authors proposed fine-grained current regulation and voltage upscaling as a circuit level technique to improve the lifetime of PCRAMs.

Domain wall memory is a recently proposed spin-based memory that can achieve much higher density than STT-MRAM, PCRAM and other emerging memory technologies [76, 77]. For this reason, it is considered to be highly promising and there have been significant efforts towards the realization of DWM [14, 78–80]. Recently, a prototype of domain wall memory array was demonstrated by IBM [20]. The potential use of DWM as shift register in re-configurable architectures was proposed in [81].

2.1.2 Architectural design and evaluation

At the architecture level, the impact of inefficient writes in STT-MRAM/PCRAM is minimized by reducing the number of write operations through suitable architectural design. One approach has been through the design of hybrid cache architectures consisting of both CMOS and STT-MRAM/PCRAM [41, 45, 82–88]. The motivation behind such an approach is to selectively direct memory blocks that incur large number of writes to CMOS memory, while storing the rest in STT-MRAM or PCRAM. Subsequently, there were several approaches that proposed suitable cache management policies like adaptive line replacement [89], to reduce the writes to STT-MRAM in a hybrid cache architecture. In [85], the authors proposed an adaptive hybrid cache architecture in which part of the cache was reconfigured as a software controlled scratch pad memory to improve energy efficiency. Another approach to reducing the write intensity in STT-MRAM based lower level cache is write-biasing, which increases the residency of dirty blocks to avoid repeated writes [90]. An alternate approach to address the write-inefficiency is to eliminate redundant writes to memory, either by comparing the data before performing the write operation [63, 91] or by tracking the dirty blocks at a finer granularity [92, 93]. In the context of multi-level STT-MRAM cache, set-remapping [66] was proposed as a technique for energy-efficient encoding of bits to multiple resistance levels. In order to address the performance implications of inefficient writes, write buffers [93, 94] and scheduling mechanisms [95] that prioritize write requests to idle cache banks have been proposed. Some of the recent efforts have proposed volatile STT-MRAM design that relaxes the non-volatility at the device level to exploit the short lifetime of data in caches and improve the write efficiency of STT-MRAM [96,97]. Application of STT-MRAM as scratchpad memory was explored in [84].

In the context of PCRAMs, limited endurance can also be addressed by reducing the write intensity [91, 98, 99] through appropriate architectural policies. In [91], the authors proposed "Data comparison write (DCW)" to avoid writing redundant data

into the memory. To improve the efficiency of DCW, "FlipNwrite" [98], a technique that increases the amount of redundant write bits was proposed. In [99], the authors investigated the data patterns through static and dynamic profiling across different applications and proposed a frequent value based PCRAM design. In this technique, the data that are frequently written to PCRAM are stored in compressed form to reduce the write intensity. The other approach to address the endurance issue is wear-leveling [44] in which the writes are spread evenly across the entire memory array.

Apart from STT-MRAM, DWM and PCRAM, other technologies like memristors and TFET-based SRAMs have also attracted interest in recent years [48–50, 100–102].

2.2 Emerging logic devices

The candidates for the next logic switch include a wide range of devices from spin-based devices to tunnel FETs (TFET) [5]. These switches are still in their nascent stages and several efforts are currently directed towards the research and development of these technologies.

Several different spintronic logic styles such as Nano-Magnetic logic (NML), All-Spin Logic (ASL), Domain Wall Logic (DWL), mLogic, *etc.* have been proposed [23, 24, 26, 28]. They use magnetization direction of the electron spin to represent binary information and different spintronic device characteristics to perform logic computation. NML uses dipolar coupling across neighboring magnets to realize logic functionalities [25, 103]. DWL exploits domain wall motion along a ferromagnetic nanowire and the shape anisotropy of the nanomagnets to realize logic gates using nanowires of various shapes [26–28]. ASL works on the principle of spin-torque switching and a spin-polarized current is used to switch between magnetization states [24, 104]. mLogic [28] utilizes the concepts of domain wall motion and tunneling magnetoresistance (TMR) of magnetic tunneling junctions (MTJ) to realize a logic switch. A key benefit of these logic styles is that the spin magnets are non-volatile and therefore,

retain their state when the power is switched off. As a result, they have near-zero leakage power consumption. Further, these logic styles are compact and can realize logic functionalities with lower device count compared to CMOS. However, they have several limitations as well. NML and DWL require external magnetic field to provide directionality and assist in the switching of nanomagnets [26, 103]. This significantly affects the technology scaling of these logic styles to smaller magnetic dimensions as it is challenging to concentrate the external magnetic field on to a magnet. Also, in NML, the required critical magnetic field increases with reduction in size of the magnets [104]. In addition, the generation of external magnetic field leads to significant energy overhead. When we consider ASL and mLogic, they are scalable to smaller feature size as the spin current decreases with scaling down of magnetic dimensions [24]. But, they have high short-circuit current that increases its power consumption. This effect is particularly severe for complex binary circuits that have high logic depth as the short-circuit power is consumed for a longer duration. Also, the spin diffusion length of spin channels is typically low and this introduces high overheads for realizing long interconnects. In addition to the above research efforts, which have focused on the realization of Boolean logic using spintronic devices , there have also been efforts to realize non-Boolean functions using domain wall magnets [29, 105].

Another potential alternate for CMOS (MOSFET transistor) at low voltages is the Tunnel FET (TFET). Compared to MOSFETs, whose switching speed is restricted by sub-threshold slope limitations, TFETs can offer higher speed at lower voltages. Several designs of TFETs using different materials ranging from III-V compounds to graphene [106–110] have demonstrated superior switching characteristics compared to MOSFETs at lower voltages. In [111, 112], the authors used one such device, Heterogeneous Tunnel FET (HTFET), to design a general purpose core for low voltage operation. The authors also proposed an energy efficient heterogeneous processor design using a combination of CMOS and TFET cores.

2.3 Thesis contributions

The major contributions of this work are different from and complementary to the earlier efforts in the following aspects:

Domain-specific RM processor using spin-based memories: Earlier research efforts primarily focused on general-purpose computing platforms. In this work, we focus on the design of domain-specific architectures using spin-based memories where the characteristics of devices can be matched to that of architecture and application. We focus on the design of a many-core processor for data intensive applications like recognition and mining (RM). We explore different design strategies and perform a systematic analysis to understand different architectural tradeoffs involved in the design of the memory hierarchy of the domain-specific processor using spin-based memories. We perform an exhaustive design space exploration, and analyze the impact of various circuit/architectural parameters of the many-core processor.

DWM-based cache design for general-purpose processors: Ours is the first effort to explore the design of on-chip memory hierarchy using DWM. While previous efforts have addressed the issues related to STT-MRAM and PCRAM, DWM poses fundamentally different challenges and tradeoffs. We propose an all-spin cache architecture in which all the levels in the cache hierarchy are designed using DWM. Our proposal enables the use of spin-based memories for designing L1 cache where they have conventionally not been used due to their high write latency/energy. We perform circuit and architectural optimizations to address the write energy/latency and the sequential access latency challenges associated with DWM.

DWM-based on-chip memory hierarchy for GPGPUs: GPUs are massively parallel architectures integrating hundreds to thousands of cores and are widely used to accelerate highly parallel, data-intensive applications. They exhibit a complex memory sub-system with multi-level cache hierarchy and many different cache structures at each level. We investigate, for the first time, the design of on-chip memory hierarchy of general-purpose graphics processing units (GPGPUs) using DWM. We

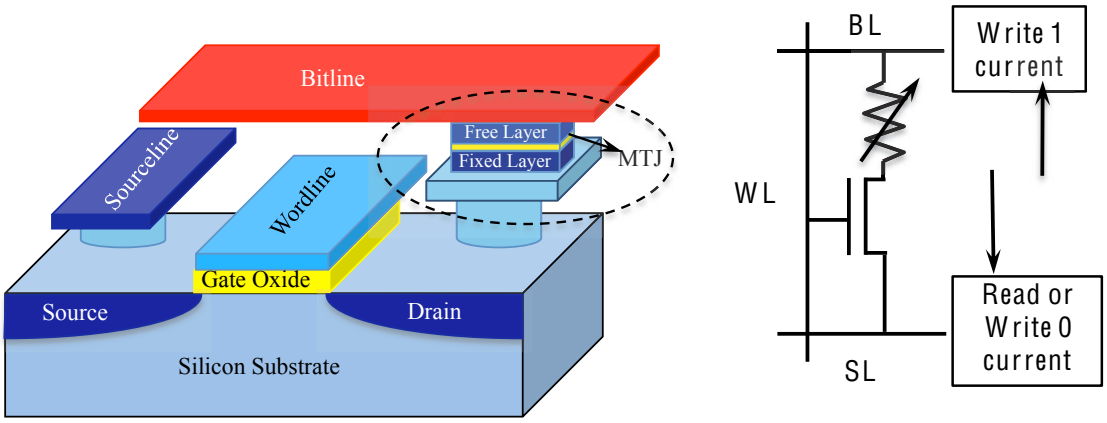
explore suitable circuit and architectural techniques that ensure that the proposed cache hierarchy benefits optimally from the high density and energy efficiency of DWM.

Stochastic logic using spintronic devices: The design of spin-based logic using technologies like All-spin logic (ASL) faces a major challenge due to their high short circuit current leading to high energy consumption. We explore a new direction that uses stochastic computing (SC) to realize logic using spin devices. SC enables compact implementation of various commonly used arithmetic functions like adders, multipliers, *etc.* with low complexity and logic depth. We show that this property of SC benefits spintronic logic, resulting in highly energy-efficient designs. Conversely, our proposal also demonstrates that SC can benefit from spintronics by exploiting the physical characteristics of spin devices that leads to efficient interface circuit designs.

3. BACKGROUND

In this chapter, we provide background information on different spintronic technologies – STT-MRAM and DWM for memory and ASL for logic – that are considered in this work.

3.1 STT-MRAM



(a) 3D Integration (b) Equivalent Circuit

Fig. 3.1.: STT-MRAM bit-cell

Figure 3.1a shows the structure of an STT-MRAM bit-cell consisting of a magnetic tunneling junction (MTJ) and an NMOS transistor integrated using 3D technology. The basic storage element in an STT-MRAM bit-cell is an MTJ, which consists of two ferromagnetic layers – a reference layer and a free layer – and a thin dielectric tunneling barrier. The magnetization of the reference layer is fixed while that of the free layer can be varied using spin-polarized current. The resistance offered by the MTJ depends on the relative magnetic orientation of the two ferromagnetic layers.

The junction resistance is low when the two layers are aligned to be parallel with each other ('0' state) and is high when the spin alignment of the two layers is anti-parallel ('1' state).

Read/write operation

Figure 3.1b shows the read/write operation in an STT-MRAM bit-cell using its equivalent circuit, which consists of a transistor and a variable resistance (1T-1R). In order to read the contents of the cell, the NMOS transistor is turned ON, a small bias voltage ($<< V_{DD}$) is applied to BL, and SL is grounded. The current that flows from BL to SL is compared to a reference value using a sense amplifier to determine the value stored in the cell. For writing a '0', the NMOS transistor is turned ON, BL is precharged to V_{DD}, and SL is grounded. If the current that flows from BL to SL is higher than the switching current of the MTJ, the magnetic orientation of the free layer changes from the anti-parallel direction to the parallel direction. In order to write a '1' to the cell, the voltage conditions are reversed.

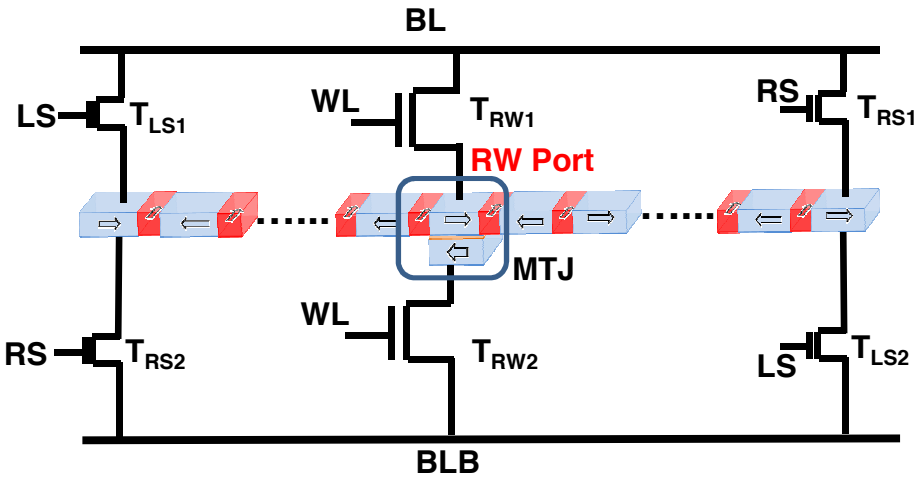
3.2 Domain Wall Memory

DWM is a spin-based memory technology that is capable of achieving very high density. In this section, we describe the design and operation of a DWM macro-cell[1] that is capable of storing multiple bits of data.

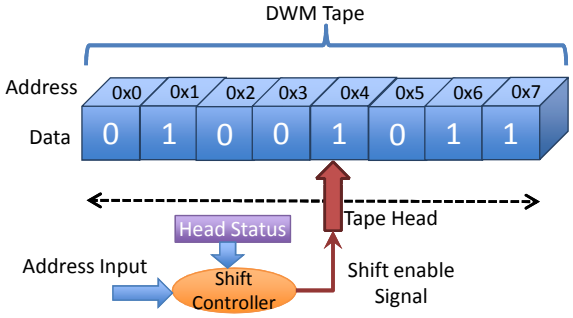
DWM macro-cell design

Figure 3.2a shows the schematic of a DWM macro-cell consisting of a ferromagnetic wire, a magnetic tunneling junction (MTJ) and access transistors. The basic storage element in a DWM macro-cell is the ferromagnetic wire. The ferromagnetic wire has multiple magnetic domains separated by domain walls. Each domain can be

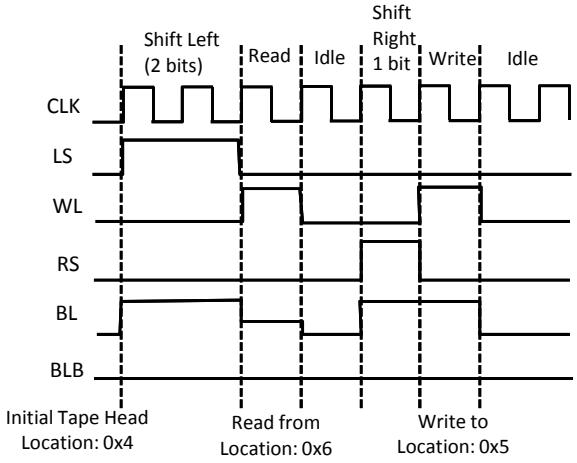
[1]We use the term DWM macro-cell to refer to a circuit that stores multiple bits per cell



(a) Schematic of a DWM macro-cell



(b) Logical view of a DWM macro-cell



(c) DWM macro-cell operation

Fig. 3.2.: DWM macro-cell

separately programmed to a certain magnetization direction, and can therefore store a single bit. Hence, a DWM macro-cell is capable of storing multiple bits of data.

DWM macro-cell operation

The read and write operations are performed using an MTJ by turning ON the corresponding access transistors (T_{RW1} and T_{RW2}) and precharging the bitlines to appropriate voltages. In order to shift the bits right, transistors T_{RS1} and T_{RS2} are

turned ON and bitline BL is connected to V_{DD} while bitline BLB is connected to GND. A left shift operation is performed by turning ON transistors T_{LS1} and T_{LS2}. This 'to and fro' movement of bits enables sharing of the read and write ports across all bits in a macro-cell, leading to very high density.

Logical view of a DWM macro-cell

Logically, a DWM macro-cell can be considered to be a tape that represents the ferromagnetic wire along with a tape head representing a read/write port, as shown in Figure 3.2b. Accessing a bit from a DWM tape involves shifting the tape head to the required location and then performing the required read/write operation. Note that tape head shifting is just a convenient logical abstraction – physically, it is the bits that move along the nanowire until the desired bit is aligned with the read/write port. Since multiple bits are stored in a macro-cell, address bits are needed to indicate which bit in the tape is being accessed. The movement of the tape head along the DWM tape is controlled by a shift controller, which determines the number of shift operations required. It does so by comparing the address bits with the current location of tape head, referred to as the head status. In order to avoid losing data during shift operations, the number of domains in the ferromagnetic wire needs to be twice the number of bits stored in the macro-cell. However, this does not incur any area penalty as, in practice, the macro-cell area is determined by the area occupied the access transistors.

Illustration of a DWM macro-cell operation

An illustration of the operation of a DWM macro-cell is shown in Figure 3.2c. Initially, the tape head is located at address 0x4. In order to read the bit stored at address 0x6, we shift the tape head to the right by 2 positions. This is done by connecting BL to V_{DD} and BLB to GND and turning ON access transistors T_{LS1} and T_{LS2} by driving the left-shift wordline LS high. Then we perform the read operation

by connecting the wordline WL to V_{DD}, bitline BL to V_{read} and bitline BLB to the GND. Suppose we then write 0 to address 0x5, which requires shifting the tape head to the left by 1 position. This is done by driving the right-shift wordline RS high to shift the bits towards the right and then connecting wordline WL to V_{DD}, bitline BL to V_{DD}, and bitline BLB to GND. For writing 1, the voltages of the bitlines would be reversed.

3.3 All-Spin Logic

All-Spin Logic (ASL) [24] is a spin-based logic design paradigm that utilizes the *non-local spin torque* exerted when a spin-polarized current is passed through a nanomagnet to influence the magnetization orientation of other connected nanomagnets. The operation of ASL is illustrated using an ASL inverter shown in Figure 3.3. It

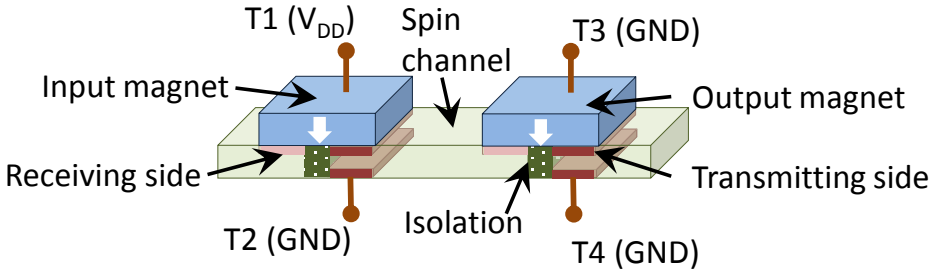


Fig. 3.3.: ASL inverter

consists of 2 magnets–an input magnet and an output magnet–that are connected through a spin channel. Let us assume that both the magnets are initially polarized in the same direction. Now, when we apply a small voltage (V_{DD}) to Terminal-1 (T1), while the other terminals (T2, T3, and T4) are connected to GND, a charge current flows from T1 to T2 through the input magnet. As a result, the electrons in the charge current whose spin orientation is parallel to the input magnet pass through, and rest with anti-parallel orientation get accumulated in the spin channel. Since there is no electron accumulation on the side of the output magnet, a *spin potential difference* is created in the channel. This causes spin current to flow through the

channel and exert a spin torque on the output magnet. The torque switches the output magnet opposite to the orientation of the input, thus realizing the functionality of an inverter. In order to achieve directionality, i.e., to avoid the output magnet from influencing the input, the channel region directly below the magnets is engineered with an isolation interface as shown in Figure 3.3. By employing the above principle and suitably connecting multiple nanomagnets, logic gates of different functionality can be realized using ASL.

A key benefit of ASL is that the spin magnets are non-volatile and has the ability to retain their previous state. A spin magnet can therefore, potentially act as a latch and this provides opportunities for fine-grained pipelined implementation of ASL. However, exploiting this opportunity would require very high levels of parallelism across inputs *i.e.*, availability of a large number of inputs without any dependencies. The traditional binary representations typically have feedback paths and input-output dependencies, which limit the scope for pipelining.

Next, when we consider the energy consumption in ASL, it stems primarily from the short circuit current that flows through the input magnet. Since the current is required to be ON until the logic is completely evaluated, the energy consumption can be quite substantial. This is further exacerbated when multiple such ASL gates are cascaded together. Notwithstanding the significant optimizations [113], such as sharing the current across multiple magnets *etc.,* ASL incurs significant energy overheads for implementing complex Boolean logic functions.

In summary, spintronic devices offer several key benefits in terms of high density, non-volatility and low leakage power. However, they also present design challenges such as (i) high latency and energy for writes in spin-based memories, (ii) variable access latency due to shifts in DWM, (iii) high short circuit power of ASL, *etc.* In the following chapters, we explore suitable circuit and architectural optimizations to address these challenges.

4. DOMAIN-SPECIFIC MANY-CORE PROCESSOR FOR RECOGNITION AND MINING

In this chapter, we explore the design of the memory hierarchy of a domain-specific processor using spin-based memory technologies. In particular, we focus on the design of spin-based many-core processor that is intended as a programmable accelerator for recognition and mining applications.

Emerging computing workloads such as Recognition, Mining, and Synthesis [40] are expected to drive the adoption of future computing platforms, presenting parallelism that can be exploited by multi-core and many-core processors. The parallelism in these workloads primarily arises from large data sets and data parallelism in the computations. The data-intensive nature of these workloads implies that demands on memory capacity and bandwidth will also increase significantly [114]. Therefore, emerging device technologies that can address the memory related challenges are of great interest.

In this chapter, we explore the benefits of designing a hybrid spin-CMOS system using spin-based on-chip memories and CMOS processor cores. The key issues to be kept in mind in any such effort are (i) How do we match the device characteristics to the architecture or vice-versa so that we best exploit the strengths of these devices? For example, for most spin-based memories, writes are much less efficient than reads. Also, DWM provides higher density than STT-MRAM but incurs high latency on random accesses due to the need for shift operations, and (ii) What are the architectural tradeoffs enabled by the use of the new devices? For example, since spin-based memories are denser than their CMOS counterparts, the area benefits that accrue from their use may be utilized in multiple ways, including having larger on-chip memories, or increasing the number of processing cores while scaling the supply

voltage to maintain performance and reduce energy. Which is the best strategy from an energy or energy-delay point of view?

The significant contributions described in this chapter are as follows:

- The design of a many-core processor for recognition and mining (RM) applications using STT-MRAM and DWM. Considering the application and architecture characteristics, we show that STT-MRAM is a suitable candidate for the second-level memory, and DWM is tailor-made for the first-level streaming memory.

- Evaluation of several design strategies and architectural tradeoffs enabled by the use of spintronic memories. We evaluate the benefits of using STT-MRAM and DWM as drop-in replacements, and also explore the possibility of investing the area savings from these replacements to increase on-chip memory capacity, and/or the degree of parallelism (by increasing the number of cores and scaling the supply voltage). We also analyze the impact of various architectural parameters on energy and performance, demonstrating that significant benefits can be achieved by synergistically exploring architectural tradeoffs along with the use of spin-based memory.

- Evaluation of the proposed RM processor design using three representative RM algorithms - Support Vector Machines (SVM), k-means clustering, and Generalized Learning Vector Quantization (GLVQ). The results indicate that spin-based memory technologies are likely to greatly benefit future data-intensive workloads such as Recognition and Mining.

The rest of this chapter is organized as follows. Section 4.1 presents the proposed many-core RM processor design. Section 4.2 describes the architectural modeling framework for the RM processor. Section 4.3 presents the experimental results and Section 4.4 concludes this chapter.

4.1 Many-core RM processor design

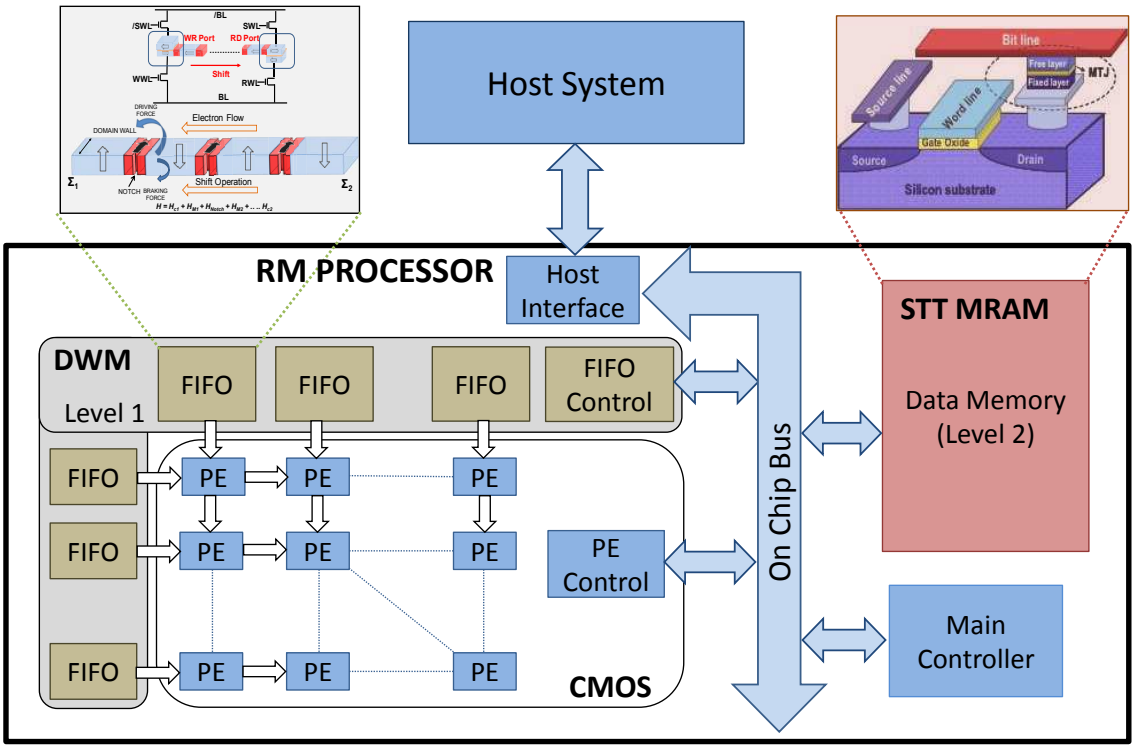


Fig. 4.1.: Many-core RM processor design

In this section, we describe the many-core RM processor design using STT-MRAM and DWM. Figure 4.1 shows the architecture of the many-core RM processor. The processor is intended as a programmable accelerator for RM applications, therefore it is specialized to efficiently execute the computational kernels that dominate these workloads. The RM processor consists of a 2-dimensional array of processing elements (PEs), two arrays of FIFOs, and a data memory. The processor implements a set of vector operations that are executed by streaming the data from the horizontal and vertical FIFOs through the PE array. An on-chip bus is used to interconnect the two levels of memory, as well as interface to the host system (*e.g.*, SoC or server) in which the RM processor resides. The host processor transfers data into the data memory, downloads the program to be executed on the RM processor, initiates execution by

writing to a special memory-mapped register, and transfers the results back into the host memory.

We next discuss the two-level memory hierarchy used in the RM processor. FIFOs, which represent the first level in the memory hierarchy, provide fast streaming access to data. The second level in the memory hierarchy is the data memory, which is of much larger size so as to store sizable parts of the data set being processed. In the baseline CMOS design, FIFOs and data memory (64KB and 2MB, respectively) represent a significant portion (roughly 75%) of the total chip area. Moreover, the leakage power of these memories contribute substantially to the total energy consumption of the RM processor.

The nature of the access characteristics to the first and second level memories in the RM processor need to be considered in order to determine the choice of memory technology. The first level memory is filled by transferring data from the second level memory over the on-chip bus in large bursts. Then, data is read out to the PE array in a streaming manner (*i.e.*, the elements are read in the order in which they were stored, one per clock cycle) for processing. In RM algorithms, it is common to require vector operations (*e.g.* dot product or Euclidean distance computation) between two large sets of vectors. To maximize data reuse, the vectors in one set of FIFOs are kept unchanged (*e.g.*, support vectors in the case of SVMs), while the vectors in the other set of FIFOs are replaced (*e.g.*, training or classification data in the case of SVM). Thus, the streaming read operation is more common than the write operation. As described earlier, DWMs are tailor-made for streaming reads of data. Therefore, we chose to use them to implement the first level memory in the RM processor.

Table 4.1.: Second level memory access characteristics for SVM and k-means

Algorithm	SVM	k-means
No. of memory reads (in bytes)	$1.02\text{x}10^{10}$	$5.6\text{x}10^{7}$
No. of memory writes (in bytes)	$6.09\text{x}10^{7}$	$4.63\text{x}10^{5}$

The second level memory in the RM processor needs to be randomly accessed with a low latency (to minimize the performance impact of data transfers to the first level memory), making DWMs less suitable. The organization of the data memory should also support a wide interface (in the baseline CMOS design, a 256-bit on-chip bus is used to the connect the two levels of memory). If we consider the nature of accesses to the second level memory (shown in Table 4.1 for the SVM and k-means algorithms), we can see that the number of read operations is greater than the number of write operations by two orders of magnitude. Finally, the leakage power of the second level memory is a major contributor to the energy consumption of the RM processor. Based on these considerations, we conclude that STT-MRAM, which has very high density and low leakage power compared to traditional CMOS-based memories while preserving fast random access capability, is a good choice for the second level memory in the RM processor. The highly read-intensive nature of the memory accesses implies that the penalty of inefficient writes into the STT-MRAM is incurred quite infrequently.

Simply performing a drop-in replacement of the CMOS memories with STT-MRAM and DWM memories may lead to improvements over the baseline CMOS design, but falls far short of the goal of optimally utilizing the potential offered by spin-based memories. In order to achieve optimum benefits, we need to re-invest the area savings to increase the number of PEs and/or on-chip memory considering various circuit/architecture tradeoffs involved in RM processor design. Figure 4.2 presents a qualitative summary of the impact of tuning different design parameters on area, performance and various components of energy consumption. Note that tuning a parameter may result in improvements in certain design metrics, while degrading others. We next discuss the architectural parameters and their associated tradeoffs in greater detail.

- Number of PEs/FIFOs: A parameter that has first-order impact on the performance and energy consumption of the RM processor is the number of PEs. The reduction in area achieved by using high density memory can be used to

	No. of PEs and FIFOs	Voltage Scaling of PEs	FIFO Depth
Performance	↑	↓	↑
PE Dynamic Energy	↑	↓	—
PE Leakage Energy	↑	↓	↓
Level-1 Memory Dynamic Energy	↓	—	↓
Level-2 Memory Dynamic Energy	↓	—	↓
Level-1 Memory Leakage Energy	↓	↑	↑
Level-2 Memory Leakage Energy	↓	↑	↓
Area	↑	—	↑

Fig. 4.2.: Impact of tuning architectural parameters on the RM processor characteristics

increase the number of PEs. In this way, we can take advantage of the inherent parallelism in the application and improve the performance of the system. Note that increase in the number of PEs should be accompanied by a corresponding increase in the number of FIFOs in our architecture (an mxn array of PEs requires $m + n$ FIFOs). When we consider the impact of increasing the number PEs/FIFOs on the energy consumption, we see that there is reduction in the energy consumed by level-1 and level-2 memory, while energy consumed by the PEs increases. This can be explained as follows: (i) Improvement in performance results in reduction in leakage energy consumed by memories. When we consider the leakage energy of PEs, the leakage energy contribution from a single PE decreases due to the improvement in performance. However, the total number of PEs also increases. The performance improvement is not propor-

tional to the increase in number of PEs. Therefore, the overall leakage energy consumed by all the PEs increases. (ii) Increasing the number of PEs increases the number of computations performed per memory access, thereby reducing the number of memory accesses required for executing an application. This reduces the dynamic energy consumed by memories. (iii) Increasing the number of PEs also increases the number of idle PEs waiting for data. This results in increased dynamic energy consumption of PEs.

- FIFO depth: A FIFO of larger depth increases data reuse, thereby improving the system performance. This improvement in performance reduces the leakage energy consumption of PEs and memory. Also due to increased data reuse, the number of write operations to the FIFOs and the number of read operations from the data memory decreases, thereby reducing the dynamic energy consumed by memories. However, the leakage energy of FIFOs would increase due to the larger FIFO size. Note that this overhead is significant in the case of CMOS-based design, but is negligible with DWM due to its inherent near-zero leakage. On the other hand, a very large FIFO would increase the energy required for shifting data in the DWM. As a result, energy required for every read/write operation would increase. Therefore, we need to use an appropriate FIFO depth considering all the above factors.

- Supply voltage of PEs: Scaling the supply voltage of PEs can be used to tradeoff energy with performance. While scaling the supply voltage of PEs leads to energy savings in the PEs, it also leads to degradation in performance. This degradation in performance causes increased leakage energy consumption from the memory. As a result, the total system energy consumption could increase if the supply voltage is scaled beyond a certain point.

Therefore, in order to design an optimal RM processor, we need to consider the complex interactions between data memory, FIFOs and PEs and perform a systematic design space exploration considering the architectural tradeoffs described above.

4.2 RM processor modeling

In this section, we describe the various steps involved in modeling and evaluation of the proposed many-core RM processor. First, we evaluated the STT-MRAM array and DWM FIFO using Spin-CACTI tool and physics-based simulations, respectively. The detailed modeling framework is described in Chapter 9. The results of these evaluations were then used to evaluate the RM processor as described below.

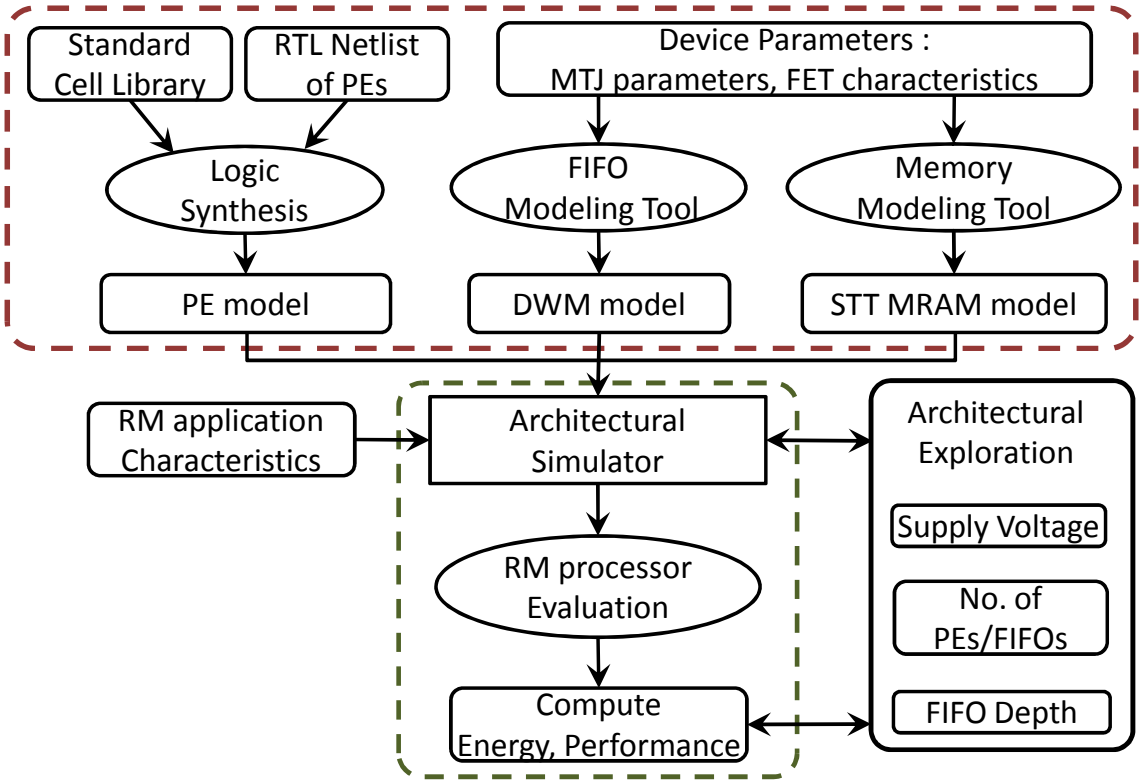


Fig. 4.3.: RM processor modeling framework

Figure 4.3 shows the modeling framework used to model the RM processor designed using hybrid technologies. The processing elements (PEs) in the RM processor were synthesized using a commercial logic synthesis tool to compute energy, delay and area. STT-MRAM memory and DWM FIFOs were modeled as described in Chapter 9. Then, we use an architecture level simulator to compute traces of the number of memory read/write operations, reads/writes to each FIFO, and the total number of

computations performed by PEs to execute an application. Based on these statistics, we evaluate the RM processor as follows:

For a RM processor with N_{FIFO} FIFOs, N_{PE} PEs, the dynamic energy consumed is computed as follows:

$$\begin{aligned} E_{dyn} &= N_{Mread}E_{Mread} + N_{Mwrite}E_{Mwrite} \\ &\quad + \sum_{i=0}^{N_{FIFO}} (N_{Fread,i}E_{Fread,i} + N_{Fwrite,i}E_{Fwrite,i}) \\ &\quad + N_{comp}E_{PE} \end{aligned} \tag{4.1}$$

where N_{Mread} is the total number of memory read operations, E_{Mread} is the energy required for a memory read operation, N_{Mwrite} is the total number of memory write operations, E_{Mwrite} is the energy required for a memory write operation, $N_{Fread,i}$ is the number of read operations from the i^{th} FIFO that has a read energy of $E_{Fread,i}$, $N_{Fwrite,i}$ is the number of write operations to the i^{th} FIFO with write energy of $E_{Fwrite,i}$, N_{comp} is the total number of computations performed in the PEs, and E_{PE} is the energy consumed by a PE per computation.

The RM processor is completely pipelined. In order to evaluate the time required to execute an application, we need to consider the memory read/write latency, and mismatch between component latencies. The total time taken for executing an application in the RM processor is computed using:

$$T_{exec} = T_{mem\_init} + max(T_{Mread}, T_{comp}, T_{Mwrite})\frac{N_{comp}}{N_{PE}} \tag{4.2}$$

where T_{mem_init} is the initial overhead involved in reading data from memory, T_{comp} is the time required to perform the N_{PE} computations by the PE, T_{Mread} is the memory read latency and T_{Mwrite} is the memory write latency.

All the CMOS-based components dissipate considerable amount of leakage power, which contributes to the total energy consumption of the chip. We compute the total energy consumption using the following equation.

$$\begin{aligned} E_{Total} &= E_{dyn} + T_{exec}(P_{Mleak} + N_{FIFO}P_{Fleak} \\ &\quad + N_{PE}P_{PEleak}) \end{aligned} \tag{4.3}$$

where P_{Mleak},P_{Fleak}, P_{PEleak} are the leakage power consumptions of the second-level memory, FIFO and PE, respectively. The total area is equal to sum of the areas occupied by memory, FIFOs and processing elements.

In this way, we evaluate the performance metrics of the RM processor and analyze the benefits of using spin-based memories. In order to understand the tradeoffs involved in the design of the RM processor, we need to perform an exhaustive exploration by varying the different parameters described in Section 4.1. For this purpose, the modeling framework was extended to allow tuning of various circuit/architectural parameters as shown in Figure 4.3. Depending on the parameter of interest, the architectural simulator performs an exhaustive exploration to obtain the optimal design point. This process was repeated for different optimization parameters and also for various application workloads to obtain the optimal processor configuration.

4.3 Experimental results

In this section, we first describe the experimental setup used to evaluate the proposed RM processor design using spin-based memory technologies. We explore the benefits of two different design strategies - drop-in replacement and iso-area re-design. Next we perform different experiments to quantify the benefits offered by individual memory technologies. We then perform design-space exploration to investigate the tradeoffs involved in tuning different circuit and architectural parameters like supply voltage, FIFO depth and number of PEs.

4.3.1 Experimental setup

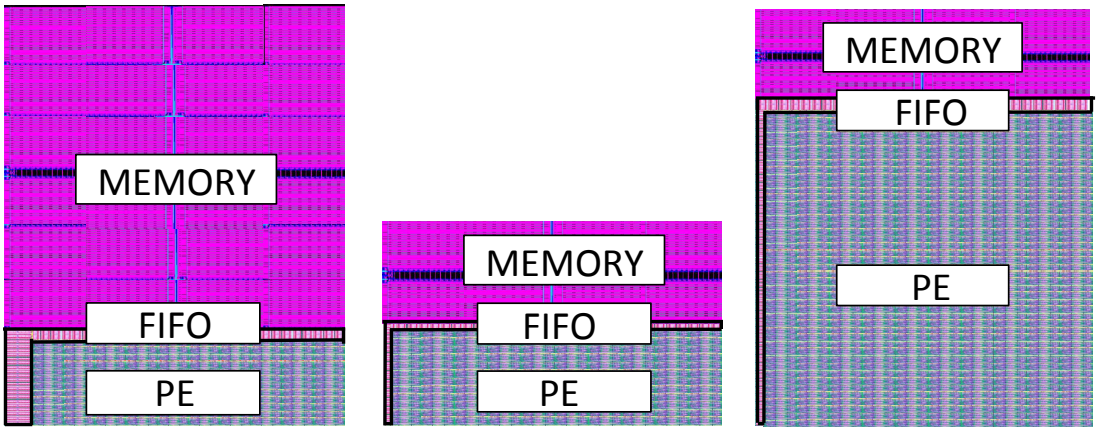
We consider a 45nm process technology in our analysis for CMOS, DWM, and STT-MRAM. The processing elements in the RM processor are implemented in IBM 45nm CMOS technology using Synopsys Design Compiler [115]. For the baseline implementation, CMOS memories are based on SRAM and are evaluated using CACTI [116]. In the STT-MRAM design, MTJ properties were chosen in accordance with [53] and we developed Spin-CACTI, a modified CACTI tool based on the model described in Chapter 9 for evaluating STT-MRAM arrays. For DWM-based FIFOs, we have developed a self-consistent simulation framework in Matlab, as explained in Chapter 9. The hybrid simulation framework can integrate the results from individual simulation environments: (a) Spin-CACTI for STT-MRAM, (b) self-consistent framework for DWM, and (c) Nanosim [117] for CMOS logic. The baseline RM processor configuration used in our analysis is provided in Table 4.2. We used the MNIST [118] and UCB [119] data sets to evaluate our design.

Table 4.2.: RM processor configuration

Data Memory Size	2MB
Number of FIFOs	64
FIFO Size	1KB
No. of PEs	1024
Bus Width	256 bits

4.3.2 RM processor evaluation

In order to evaluate the proposed spin memory based RM processor, we consider three different RM algorithms - Support Vector Machines (SVM) [120], k-means clustering [121] and Generalized Learning Vector Quantization (GLVQ) [122]. The RM processor was evaluated under two design strategies: drop-in replacement and iso-area re-design, which are illustrated in Figure 4.4. In drop-in replacement, we replace



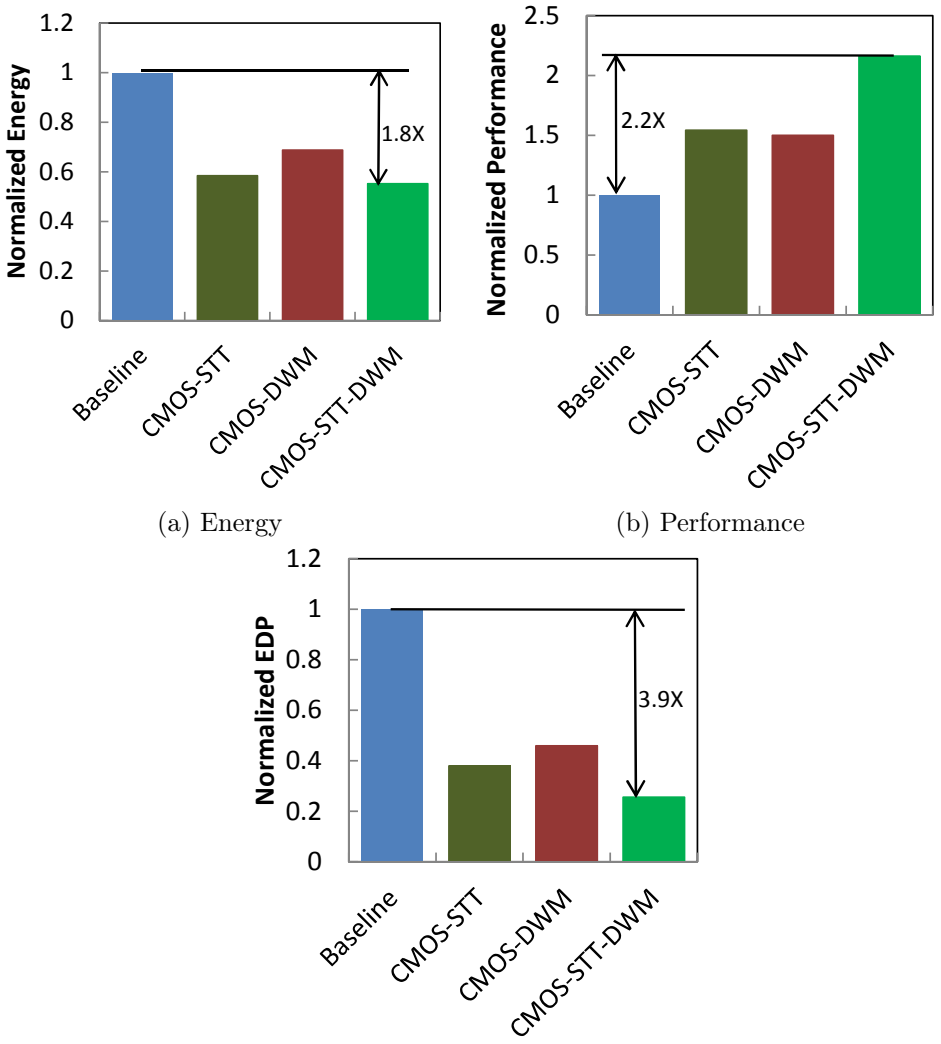
(a) Baseline CMOS (b) Drop-in replacement (c) Iso-area re-design

Fig. 4.4.: Spin-based RM processor: Design strategies

the CMOS memories with corresponding spin memories of the same capacity, while keeping all the architectural parameters constant. In iso-area re-design, we re-invest the area benefits that result from the use of high density spin memories to increase the number of PEs and the FIFO size and scale the supply voltage to obtain the design with optimum energy-delay product. We evaluate these designs in terms of both total energy consumed and performance measured in terms of the number of operations per second. An operation is defined as a dot-product computation for SVM, and distance computation for k-means and GLVQ. The effectiveness of these approaches is analyzed in the following sections.

Table 4.3.: Comparison of the spin-based design of the RM processor with the baseline implementation

Algorithm	Baseline			Drop-in replacement			Iso-area re-design		
	Area (mm^2)	Energy (mJ)	Performance (GOPS)	Area (mm^2)	Energy (mJ)	Performance (GOPS)	Area (mm^2)	Energy (mJ)	Performance (GOPS)
SVM	7.84	4.93	285	2.40	4.54	224	7.49	2.72	616
k-means		0.11	53		0.10	53		0.10	78
GLVQ		0.048	309		0.045	309		0.039	398



(a) Energy

(b) Performance

(c) Energy-Delay Product

Fig. 4.5.: Energy, Performance, and Energy-Delay Product comparison to analyze the benefits of DWM and STT-MRAM

Drop-in replacement

Table 4.3 presents results comparing the spin-based RM processor design with the baseline CMOS design. In the case of the drop-in replacement approach, there is considerable area improvement due to the high density of STT-MRAM and DWM. Drop-in replacement also results in a small reduction in energy for all the three algorithms, which can be attributed to the improvements in leakage energy and read

energy of spin-based memories. Although the write energy of the spin-based memories is higher than CMOS-based memories, this does not significantly affect the overall energy as reads greatly outnumber writes for the applications under consideration. The performance of the RM processor does not change in the case of k-means and GLVQ algorithms. This is because, the improvement in memory read latency obtained by using STT-MRAM gets nullified by the increased write latency of the DWM FIFO. The degradation in performance observed in the case of the SVM algorithm can be attributed to the higher amount of data written from the PEs to the STT-MRAM.

Iso-area re-design

In the case of the iso-area re-design strategy, the reduction in area obtained by using high density STT-MRAM and DWM is re-invested to increase the number of PEs and the FIFO size. As shown in Table 4.3, iso-area re-design results in both performance and energy benefits compared to drop-in replacement. The improvement in performance is mainly due to the increase in number of PEs, which helps due to the highly parallelizable nature of RM applications. The other factor that helps to improve the performance is the large FIFO size. This helps to burst read a larger amount of data from the memory, thereby improving the efficiency of data transfer between the two levels of memory. This reduces the number of cycles for which PEs are stalled waiting for data. In addition, large FIFO size enables efficient data re-use, which reduces the number of read operations from STT-MRAM and write operations to DWM. The reduction in energy consumption is achieved in three different components: 1) leakage energy, 2) write energy of FIFOs, and 3) read energy of the second level memory.

4.3.3 Analysis of benefits of DWM and STT-MRAM

For a systematic analysis of benefits and drawbacks of each technology (DWM and STT-MRAM), we consider four different designs of the RM processor under iso-area conditions:

1. A design using only CMOS (Baseline),
2. A design in which SRAM data memory is replaced with STT-MRAM memory (CMOS-STT),
3. A design in which CMOS FIFOs is replaced with DWM FIFOs (CMOS-DWM),
4. A design, which uses both DWM FIFOs and STT-MRAM memory (CMOS-STT-DWM).

We compare the energy, performance and energy-delay product of the above designs in Figure 4.5 for the SVM algorithm.

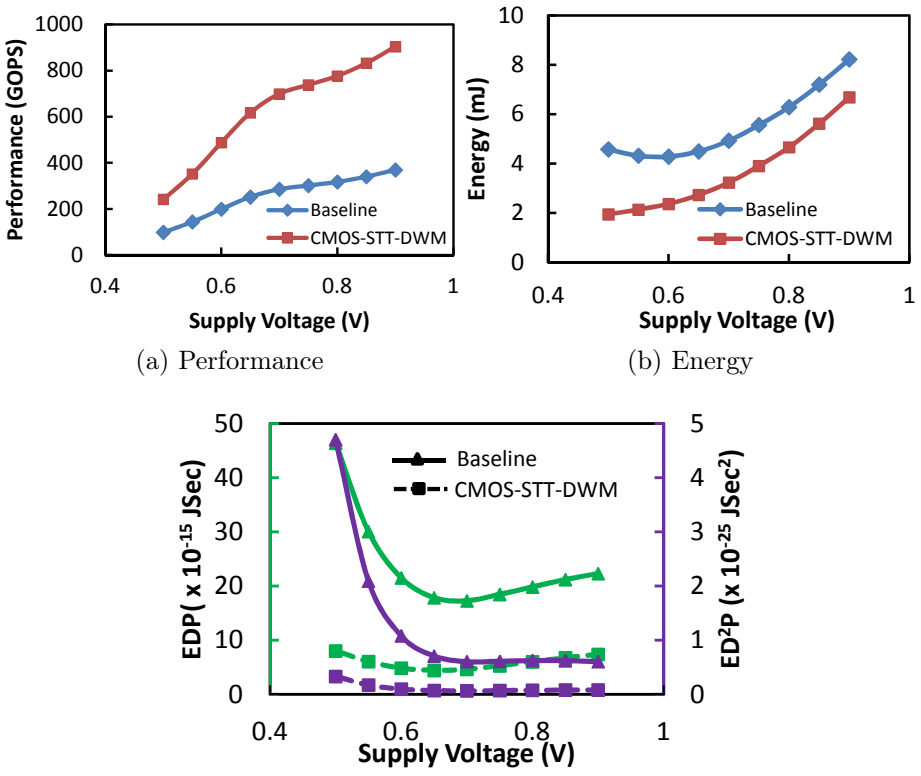
In the CMOS-STT design, we replace the SRAM data memory with a STT-MRAM memory of the same capacity and increase the number of PEs. In this design, memory read latency, memory read energy and leakage energy are reduced compared to the CMOS baseline, while the write energy and write latency of memory increase. Since writes are very infrequent in this application, we see an overall improvement in the performance and energy of the RM processor.

In the CMOS-DWM design, we replace CMOS FIFOs in the baseline implementation with DWM FIFOs of larger size. In this case, there is a reduction in the leakage power, FIFO read energy, and FIFO read latency. On the other hand, the write energy and write latency of the FIFO increase. However, the FIFO write latency is hidden by the high read latency of SRAM memory and therefore, does not affect the performance of the RM processor. The improvement in performance, as shown in Figure 4.5b, is due to the decrease in PE stall cycles resulting from larger FIFOs. As we will see in Section 4.3.5, larger FIFOs also improve the data reuse, thereby reducing the number of memory read operations. This also causes a reduction in

the number of FIFO write operations, which improves the energy consumption and performance of the RM processor.

The CMOS-STT-DWM design combines the benefits of both the CMOS-STT and CMOS-DWM designs and therefore, performs superior to all the other designs. We obtain 1.8X reduction in energy 2.2X improvement in performance, and 3.9X reduction in energy-delay product in the CMOS-STT-DWM design compared to the CMOS baseline implementation. In rest of the chapter, we will focus on exploring circuit and architecture optimizations for the CMOS-STT-DWM design.

4.3.4 Circuit optimization: Voltage scaling

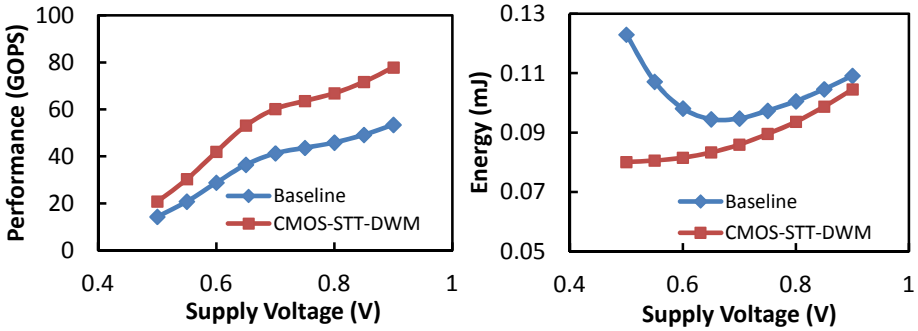


(a) Performance

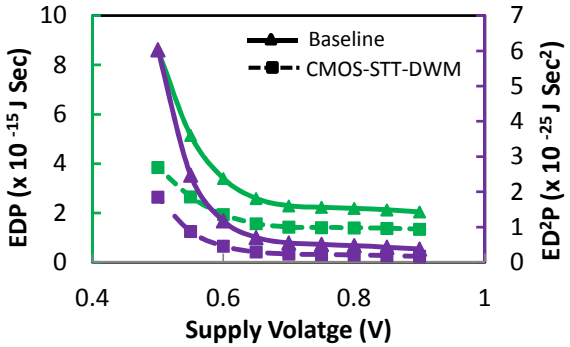
(b) Energy

(c) EDP and ED^{2}P comparison

Fig. 4.6.: Effect of voltage scaling for SVM algorithm



(a) Performance (b) Energy



(c) EDP and ED^{2}P comparison

Fig. 4.7.: Effect of voltage scaling for k-means algorithm

In this section, we analyze the impact of scaling the supply voltage of PEs in our CMOS-STT-DWM design and compare it with the baseline CMOS-based RM processor for the SVM and k-means algorithms. Scaling of supply voltage causes the clock period of the RM processor to increase, resulting in degradation in performance of both the CMOS baseline and CMOS-STT-DWM designs as shown in Figure 4.6a and 4.7a. Also note that the degradation in performance is higher at lower voltages due to the non-linear relationship between the gate delay and supply voltage.

The impact of voltage scaling on the energy consumption for SVM and k-means applications is shown in Figure 4.6b and 4.7b respectively. As we scale the voltage, the energy consumption of the PEs decreases. However, due to increase in the execution time, the duration for which the memory stays idle also increases, resulting in increased leakage energy. As the supply voltage is scaled beyond a certain threshold,

leakage energy of the memory begins to dominate the PE energy, causing an increase in the overall energy consumption of the system. The energy-optimal voltage is determined by percentage of total energy that is contributed by leakage energy of memory. Higher the leakage energy contribution, lower is the amount of voltage scaling possible. Since the leakage power in the case of spin-based memories is much lower than CMOS-based memories, leakage energy of memory does not play a significant role in total energy of the CMOS-STT-DWM design. Therefore, the supply voltage of the CMOS-STT-DWM design can be scaled further when compared to the baseline to obtain larger benefits in energy.

In order to demonstrate the overall improvement of the system in terms of both performance and energy, we consider Energy-Delay Product (EDP) and Energy-Delay^2 Product (ED^{2}P) and illustrate the results for the SVM and k-means algorithms. As shown in Figure 4.6c and 4.7c, the spin-based RM processor can be scaled more compared to the CMOS-based RM processor to obtain energy benefits with negligible performance degradation. Also, note that the optimal operating point depends on the application. The contribution of dynamic energy consumed by PEs to the total energy consumption is much higher in the case of SVM algorithm compared to k-means algorithm. This enables us to scale the supply voltage more for SVM compared to k-means.

4.3.5 Architectural exploration

In this section, we study the different architectural tradeoffs involved in the design of the RM processor using spin-based memories.

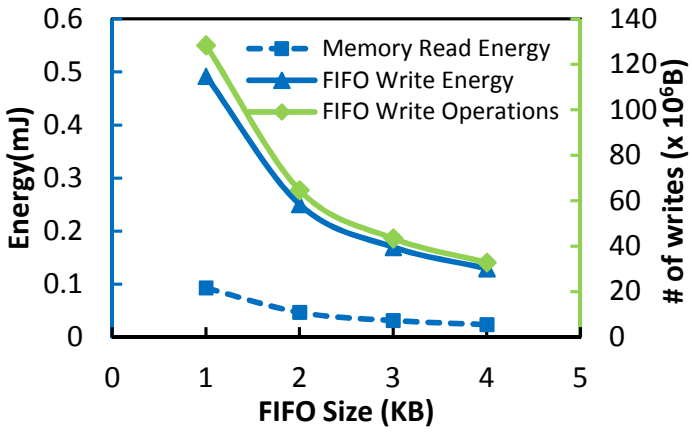
FIFO size

As discussed in Chapter 3, write energy is one of the major concerns with STT-MRAM and DWM. We have seen that RM applications have a much larger number of reads than writes. This makes STT-MRAM a natural choice for such applications.

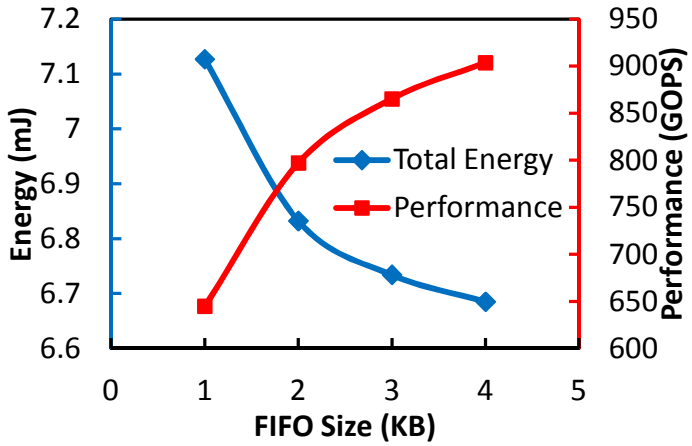
However, higher write energy of FIFOs is a major bottleneck, underscoring the need for architectural optimization.

In order to analyze the impact of FIFO sizing, we consider the number of FIFO write operations required as a function of FIFO size. Figure 4.8a shows that the number of FIFO write operations decrease as the FIFO size is increased. This can be explained as follows. Consider an SVM application with $n1$ support vectors and $n2$ test vectors, each with dimensionality 'd', and a RM processor with a PE array of size pxp and FIFO depth 'D'. We need to compute the dot product between every test vector and support vector pair. In typical SVM applications, $n1$ and $n2$ are very large, typically much larger than p. Let us assume for a moment that $D = d$, which implies that only one support/test vector can be stored in a FIFO at a time. The RM processor may retain support vectors in one set of FIFOs while cycling through the test vectors in the other set, or vice versa. Let us consider the scenario where support vectors are retained in the FIFOs to perform dot product with every test vector. When a test vector is written to the FIFO once, we can perform a dot product between this test vector and 'p' support vectors. This would require every test vector to be written $n1/p$ times to the FIFO to perform dot products with all the support vectors. If the depth of the FIFOs is increased to $D = kd$, each FIFO can hold 'k' vectors. When a test vector is written to the FIFO, the dot product can be performed between this test vector and 'kp' support vectors. As a result, the number of FIFO write operations decreases by a factor of 'k'.

The decrease in number of FIFO writes also implies that the number of read operations from the data memory is reduced. As a result, there is reduction in FIFO write energy and data memory read energy as shown in Figure 4.8a. The data reuse offered by increased FIFO size results in a reduction in total energy consumption and improvement in performance of the RM processor, as shown in Figure 4.8b. However, increasing the size of FIFO increases the number of shifts required per read/write operation. Therefore, every read/write operation would now require higher energy



(a)



(b)

Fig. 4.8.: Effect of FIFO size on energy consumption for SVM algorithm

per access. Also, increasing the size of FIFO increases the failure probability of DWM [14], which needs to be taken into account during the design process.

Number of PEs

In this section, we explore the implications of increasing the number of PEs on energy and performance of the RM processor. In this experiment, as the number of PEs increases, FIFO count is increased accordingly (the number of PEs is the square of half the number of FIFOs). All the other parameters are kept constant.

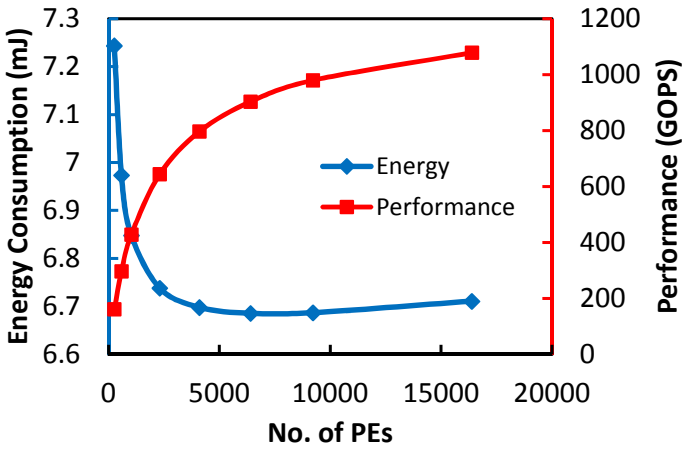


Fig. 4.9.: Effect of increasing the no. of PEs on energy and performance for SVM algorithm

Figure 4.9 shows the impact on energy and performance of the processor for the SVM application. The performance of the system initially increases with increasing number of PEs due to the highly parallel nature of the workload. However, as we increase the number of PEs beyond a certain threshold, the system performance is constrained by the memory bandwidth. As a result, no further improvement in performance is possible without optimizing the memory.

In order to understand the impact of number of PEs on energy, let us consider the different components of energy consumption:

- **FIFO write and memory read energies**: As discussed in Section 4.3.5, the number of FIFO write operations is $n1/p$. When we increase the number of PEs, p increases, which reduces the FIFO write energy and memory read energy.

- **FIFO read energy**: Increasing the number of PEs also increases the number of computations being performed per FIFO read. Therefore, the total FIFO read energy is also reduced.

- **PE dynamic energy**: Given an application, the total number of computations remains the same. Hence, the total dynamic energy of the PEs remains roughly constant.

- **Leakage energy**: Improvement in performance also implies that leakage energy from all the components reduces. Therefore, the total energy consumption of the processor reduces as the number of PEs increases.

- **PE idle energy**: PE idle energy is the energy consumed in the PEs, which are waiting for the data from FIFOs. This occurs during the initial and final phases of a streaming computation (as the pipeline is being setup / drained). As we increase the array size, the number of idle PEs increases, resulting in increased PE idle energy.

As shown in Figure 4.9, considerable energy benefit can be obtained by increasing the number of PEs. However, as the PE idle energy begins to dominate, the total energy consumption of the RM processor increases, resulting in an optimal design point.

In summary, the results demonstrate considerable improvements in energy, performance, and energy-delay product when using spin-based memory compared to the CMOS baseline. The results also demonstrate that tuning the circuit and architectural parameters can result in significant further improvements in energy and performance.

4.4 Conclusion

We presented the design and evaluation of a many-core Recognition and Mining processor using spin-based memory technologies. We developed models of STT-MRAM and DWM memories, which were used to perform an evaluation of, and architectural exploration for, the RM processor. Our results demonstrate that spin-based memory has great potential in improving the performance of Recognition and Mining, and perhaps other parallel data-intensive workloads.

5. TAPESTRI: DESIGN OF DWM TAPES WITH SHIFT-BASED WRITE

As described in earlier chapters, the spin-based memories offer considerable benefits in terms of density and leakage power. However, the use of MTJ to perform writes in STT-MRAM and DWM results in high write energy/latency. Our analysis indicates that an 1MB STT-MRAM cache requires 1.8X more write energy and 3.5X more write time compared to an SRAM cache of the same capacity. Further, the high write current requirement of MTJ-based write demands the use of large access transistors that compromises the density benefits, and aggravates the possibility of dielectric breakdown leading to reliability concerns. In addition, the conflicting requirements of read and write operations imposes stringent design constraints, resulting in reduced stability and increased read/write failures under variations.

Many previous efforts have proposed various optimizations of MTJ-based writes, including different genres of STT-MRAM, hybrid caches, volatile STT-MRAM design *etc.* as described in Chapter 2. Although these proposals have resulted in notable improvements, there is still a significant gap between the write energy and latency of SRAM and spin-based memory.

In this chapter, we bring a completely new and different insight to address the challenge of write energy and latency in spintronic memory design — *domain wall motion, which was originally proposed for performing shift operations in DWM, offers a fast, energy-efficient alternative for performing writes.* The concept of domain wall motion is fundamentally different from the MTJ-based write mechanism used in both STT-MRAM and traditional DWM designs. This write mechanism using domain wall motion has been experimentally demonstrated to be more efficient than MTJ-based write in terms of energy and latency, as well as scalability for nanoscale magnets with perpendicular magnetic anisotropy (PMA) [21].

Our proposal, which we call TAPESTRI (TAPE with ShifT based wRIte), leverages the above insight to achieve fast, energy-efficient write operations in spintronic memories. We propose the design of two different bit-cells, 1bitDWM and MultibitDWM, which are optimized for the differing requirements of the different levels of the cache hierarchy.

- 1bitDWM is a bit-cell that is designed to optimize performance. It retains all the benefits of STT-MRAM and can match SRAM in write efficiency. Moreover, unlike conventional DWM bit-cells, it does not require any shift operations. This allows it to be used in L1 cache, where spin memories have conventionally not been used due to the high write latency/energy.

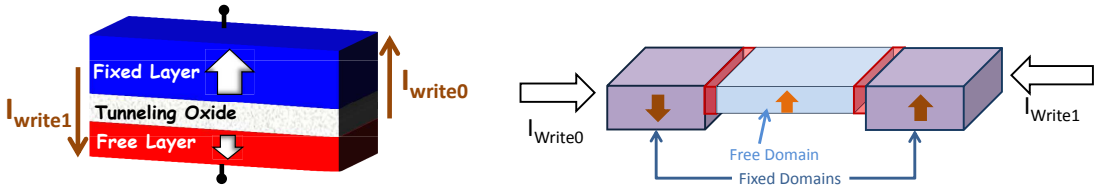
- MultibitDWM is a bit-cell that is designed to maximally utilize the density benefits of domain wall memory. It achieves much higher density than STT-MRAM (and 1bitDWM) by storing multiple bits in a single cell. However, this design, in general, requires shift operations to be performed on a cell before read/write accesses.

The proposed bit-cells also feature decoupled read-write paths that use MTJ for reads, and shifts for writes. This enables independent optimizations of read and write operations resulting in improved read/write stability along with fast, energy-efficient read and write operations. Also, the access transistors in 1bitDWM can be minimum-sized, which enables it to achieve density benefits similar to 1T-1R STT-MRAM bit-cell.

The rest of this chapter is organized as follows. Section 5.1 describes the shift-based write mechanism. Section 5.2 presents the proposed 1bitDWM and Multi-bitDWM designs. In Section 5.3, we present a comparison of the characteristics of standalone caches designed using the proposed bit-cells with SRAM and STT-MRAM and Section 5.4 concludes the chapter.

5.1 Shift-based write

The write operation in spintronic memories is typically performed by injecting current into an MTJ, which causes switching of nanomagnets through a mechanism called Spin-Transfer Torque (STT) as shown in Figure 5.1a. In order to achieve successful write operation, a current of appropriate magnitude needs to be passed through the MTJ for sufficient duration. This write mechanism leads to high write energy and write latency, which is a major challenge for designing spin-based memories.



(a) MTJ-based write (b) Shift-based write

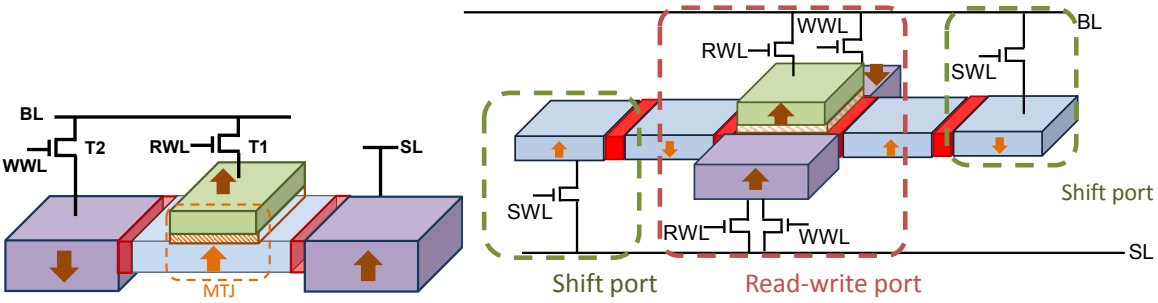
Fig. 5.1.: Different write mechanisms in spintronic memories

However, a recent development in DWM technology [21] has eliminated this inefficiency. It has been experimentally shown that domain wall motion can also be used to perform fast, energy-efficient writes in DWMs. This property, often referred as shift-based writes, is demonstrated in Figure 5.1b. The structure for write operations consists of a ferromagnetic wire with three domains – two fixed domains and a free domain. The magnetization of the two fixed domains are set to up-spin and down-spin during fabrication. However, the magnetization of the free domain, which is sandwiched between the fixed domains, can be varied by *shifting* the magnetization of one of the fixed domains by applying a current pulse in the appropriate direction.

5.2 TAPESTRI bit-cell designs

In this section, we describe the design of two different DWM bit-cells – 1bitDWM and multibitDWM – that are proposed in this work. The 1bitDWM bit-cell is optimized for latency; as the name indicates, is capable of storing only one bit per cell,

and therefore compromises on density. On the other hand, the multibitDWM bit-cell harnesses the density benefits offered by DWM by storing multiple bits in the device. In doing so, it incurs performance penalty due to shift operations[1]. In this section, we present a detailed description of these two bit-cell designs.



(a) 1bitDWM (b) MultibitDWM

Fig. 5.2.: Schematic of TAPESTRI bit-cells

5.2.1 1bitDWM

Figure 5.2a shows the schematic view of 1bitDWM bit-cell. It consists of a ferromagnetic wire with two fixed domains and a free domain, an MTJ and 2 access transistors. The data stored in the bit-cell is determined by the magnetic orientation of the free domain. When the magnetic orientation of the free domain is parallel to that of the MTJ fixed layer, then the MTJ offers low resistance indicating '0' state. When the magnetic orientation is in the opposite direction, MTJ offers high resistance indicating '1' state. The bit-cell also consists of two separate access transistors – read access transistor (T1) and write access transistor (T2), which are used to control the direction of currents during the read/write operations.

Read/write operation: Data stored in the bit-cell can be read using the MTJ and the write operation is carried-out by shifting the magnetic orientation of the ap-

[1]Note that the tradeoff between latency and density is not binary; as explored in later sections, the multibitDWM bit-cell may be configured to multiple design points by varying the number of bits (or domains) per bit-cell.

propriate fixed domain into the free domain. The voltage conditions for each signal in the bit-cell during read/write operations in 1bitDWM bit-cell are shown in Table 5.1. In order to read the contents of the cell, the read access transistor (T1) is turned ON and the bitline BL is driven high and the sourceline is grounded. The current that flows from BL to SL varies depending on the resistance offered by MTJ, which is used to determine the value stored in the cell. The write operation is performed by turning the write access transistor (T2) ON and injecting current along the ferromagnetic wire in the appropriate direction. In order to write 0, bitline BL is driven high and SL is connected to GND. This results in left shift operation, thereby writing 0 into the bit-cell. For writing 1, the voltage conditions of the bitlines are reversed.

Table 5.1.: 1bitDWM bit-cell operation

	RWL	**WWL**	**BL**	**SL**
Read	V_{DD}	0	V_{read}	0
Write 0	0	V_{DD}	V_{write}	0
Write 1	0	V_{DD}	0	V_{write}
Idle	0	0	0	0

5.2.2 MultibitDWM

The schematic of multibitDWM is shown in Figure 5.2b. It consists of two ferromagnetic wires, one MTJ and 6 access transistors. The two ferromagnetic wires are aligned orthogonal to each other with a shared magnetic domain (SMD) at their intersection. One of the ferromagnetic wires contains multiple free magnetic domains, which are used to store data. The other ferromagnetic wire consists of two fixed domains of opposite magnetic orientations and one free domain that is formed by the SMD. This allows data to be written into the SMD by using shift-based writes, realizing the write port of the bit-cell. Note that such structures have been proposed and prototyped in the context of domain wall logic [26]. In this work, we use it to achieve high density and efficient write operation simultaneously. The read port is

formed by the MTJ, similar to the 1bitDWM cell. The read and write ports require one and two access transistors respectively. In addition, the multibitDWM has shift ports, formed by 2 access transistors on the extreme left and right, that is used to shift the data bits before they are accessed.

Read/write/shift operation: Three kinds of operations can be performed in multibitDWM bit-cell – Read, write and shift. The voltage conditions of the various signals in the bit-cell used for performing the read, write and shift operations are shown in Table 5.2. Reading/writing of data to the domain at the read/write port is

Table 5.2.: MultibitDWM bit-cell operation

	RWL	**WWL**	**SWL**	**BL**	**SL**
Read	V_{DD}	0	0	V_{read}	0
Write 0	0	V_{DD}	0	V_{write}	0
Write 1	0	V_{DD}	0	0	V_{write}
Shift Left	0	0	V_{DD}	0	V_{shift}
Shift Right	0	0	V_{DD}	V_{shift}	0
Idle	0	0	0	0	0

performed in a manner similar to a 1bitDWM bit-cell described above. Shifting of bits in multibitDWM bit-cell is accomplished by turning ON the shift access transistors and precharging the bitlines to appropriate voltages. For shifting the bits towards right, BL is driven high and SL is grounded. For shifting in opposite direction, the voltage conditions of BL and SL are reversed.

5.2.3 Tapestri bit-cell characteristics

In this section, we present a detailed analysis of various key metrics and optimizations for the proposed bit-cells. We performed all the evaluations in this section at 32nm technology node. The device parameters used in the evaluation are shown in Table 5.3, unless stated otherwise.

Optimized write: One of the key features of the proposed design is that domain wall shift is used for performing writes. Figure 5.3 compares the characteristics of

Table 5.3.: DWM device dimensions (W,L, T denote the width, length and thickness of the nanomagnets, respectively. t_{ox} is oxide thickness)

DWM wire dimensions		
L (nm)	W (nm)	T (nm)
64	32	3
MTJ dimensions		
L (nm)	W (nm)	t_{ox} (nm)
64	32	1.6

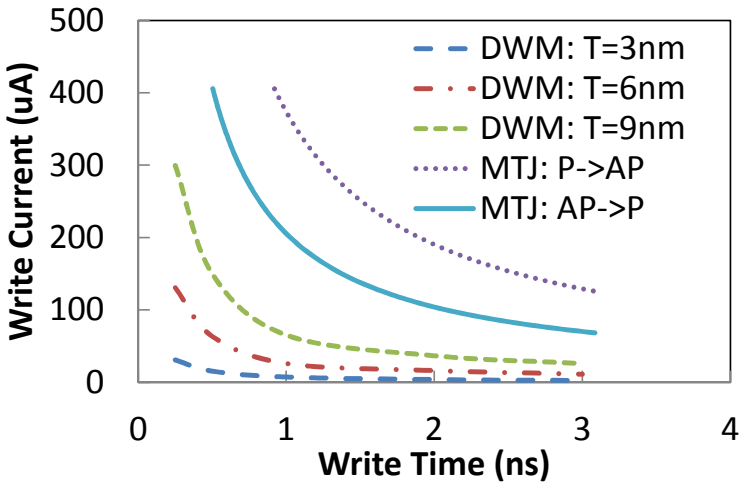
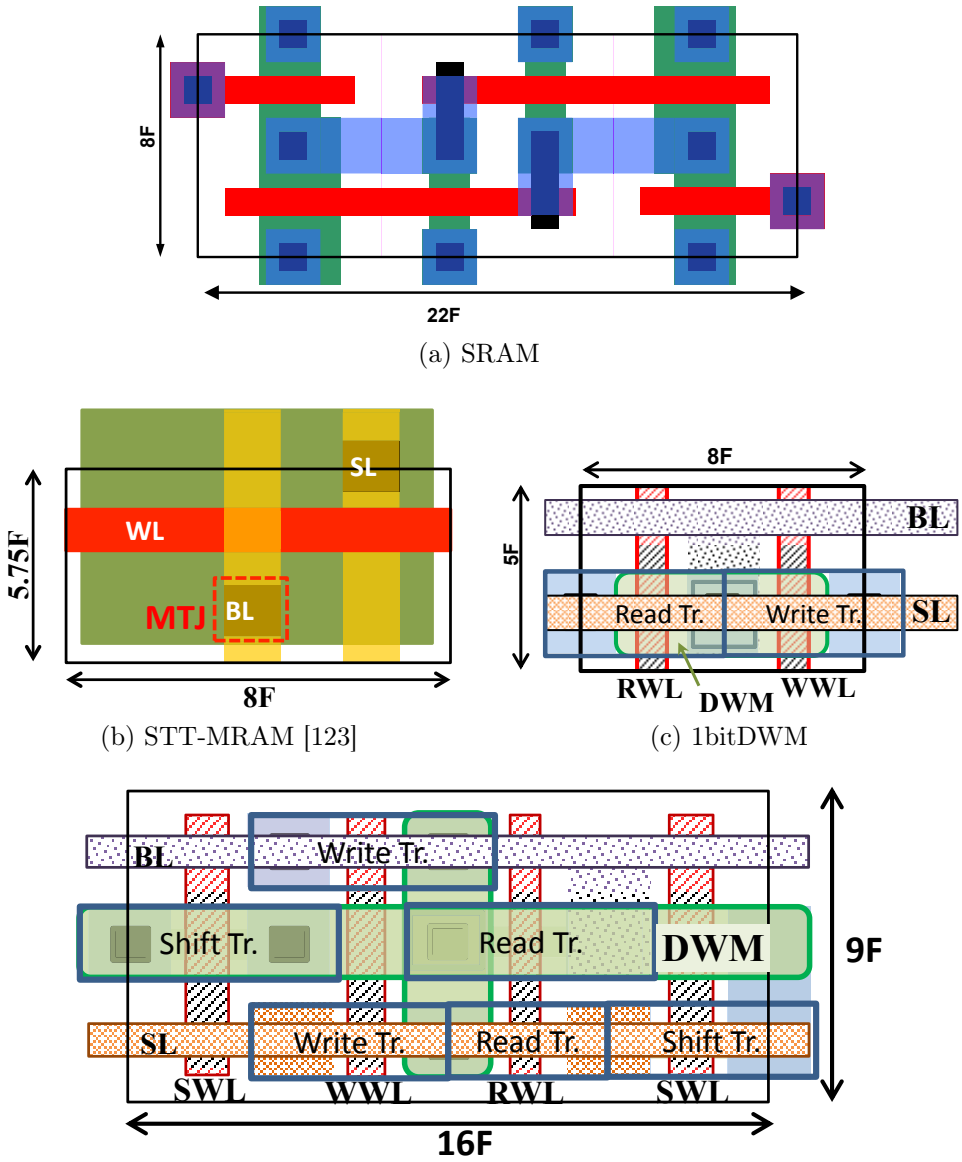


Fig. 5.3.: Comparison of write characteristics of shift-based write with MTJ-based write

domain wall motion based write with that of MTJ-based write operation for different values of ferromagnetic wire thickness (T). We consider ferromagnets with perpendicular magnetic anisotropy (PMA) in this analysis. For MTJ-based write, we considered an optimized MTJ having magnet dimensions of 32nm x 64nm and oxide thickness of 1.1nm. As we can see from the figure, domain wall motion based write has lesser current and latency requirement compared to both parallel to anti-parallel switching ($P \rightarrow AP$) and anti-parallel to parallel switching ($AP \rightarrow P$) of MTJ. This improves the efficiency of write operation in the proposed design in terms of both latency as well as energy consumption.



(a) SRAM

(b) STT-MRAM [123]

(c) 1bitDWM

(d) MultibitDWM

Fig. 5.4.: Layout of STT-MRAM, 1bitDWM, and MultibitDWM bit-cells

Density: Figure 5.4 shows the layout of SRAM, STT-MRAM, 1bitDWM and multibitDWM bit-cells. Note that the ferromagnets and MTJs are stacked on the top of the access transistors in a 3D fashion. The area of all the spin memory bit-cells is determined by the access transistors as the dimensions of ferromagnets and MTJ are relatively insignificant. As we can see from the figure, 1bitDWM bit-cell ($40F^2$) achieves $\sim 4X$ lower area than an SRAM bit-cell. Also, the area of a 1bitDWM

bit-cell is comparable to that of an STT-MRAM bit-cell ($46F^2$), despite the fact that the former has two access transistors compared to one in the later. This is because the domain wall motion based write mechanism reduces the write current requirement of the proposed bit-cell considerably enabling the use of a minimum-sized write access transistor. As a result, both the access transistors in 1bitDWM bit-cell can be minimum-sized. In comparison, SRAM uses 6 access transistors and STT-MRAM requires a large access transistors to supply the required write current. Figure 5.4 also shows that multibitDWM can achieve even higher densities compared to SRAM, STT-MRAM and 1bitDWM. A multibitDWM bit-cell storing 32 bits of data requires $4.5F^2/bit$, achieving $\sim 39X$ lower area than SRAM and $\sim 10X$ reduction over STT-MRAM.

Read optimization: In a typical 1T-1R STT-MRAM bit-cell, both read and write operations are performed using an MTJ, resulting in conflicting design requirements. For example, high Tunneling Magneto-Resistance (TMR) is required for high read stability, which requires thicker tunneling oxide. However, this significantly increases the voltage required for performing writes. Similarly, fast read operation demands for large read voltage, but it increases the chances of read disturb failure.

In the proposed bit-cells, decoupling of read/write paths enables us to design a robust bit-cell with higher oxide thickness having higher TMR and read disturb margins. Figure 5.5 shows that we can design the bit-cells over a wide range of delay constraints without any degradation in the read disturb margin. Higher oxide thickness also enables the use of voltage-mode sensing for faster read operation compared to current-mode sensing scheme that is typically used in standard STT-MRAM designs.

Reliability: One of the key advantages of the proposed design is the mitigation of reliability issues related to tunnel oxide breakdown. In standard STT-MRAM bit-cell, the write speed is mainly limited by Time-Dependent Dielectric Breakdown (TDDB) of tunneling oxide. With higher speed, the required write current density of MTJ increases, which exponentially degrades the TDDB-limited MTJ lifetime [124].

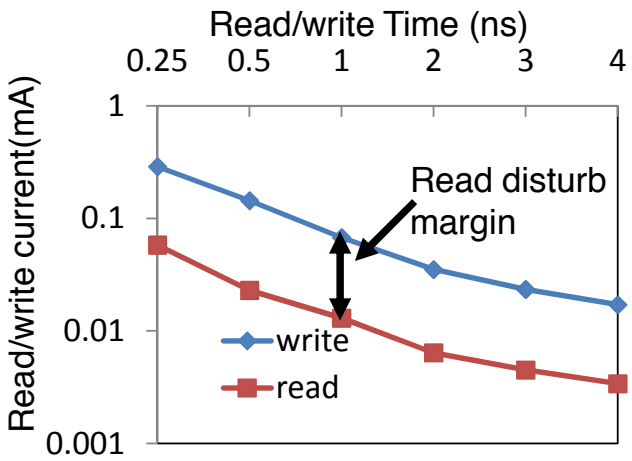


Fig. 5.5.: Read/write stability of TAPESTRI bit-cells

In the proposed design, decoupled read/write paths facilitate faster write operations without such reliability concerns.

5.3 Cache characteristics

In order to understand the tradeoffs involved in the design of caches using the proposed bit-cells, we present a comparison of the cache characteristics designed using different technologies in Figure 5.6. For this analysis, we evaluated an 128KB cache

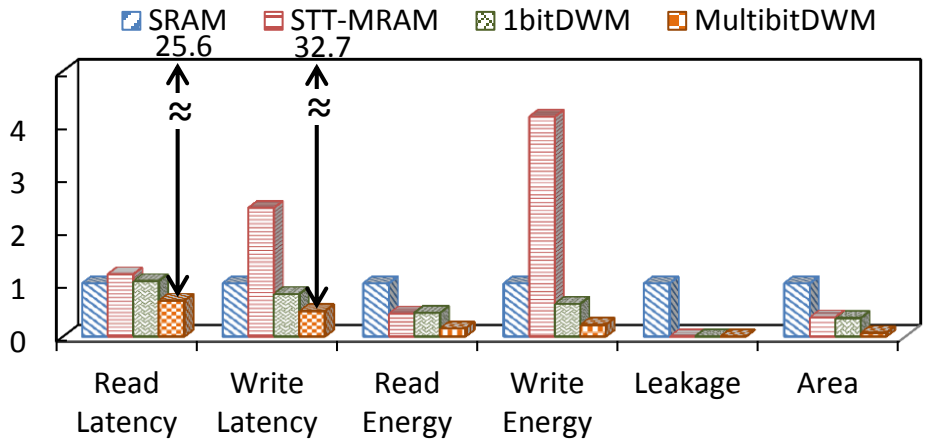


Fig. 5.6.: Comparison of DWM characteristics with SRAM and STT-MRAM

using CACTI [116] for SRAM and in-house enhanced versions of CACTI (described in Chapter 9) for STT-MRAM and DWM. For multibitDWM, we consider a bit-cell that stores 32 bits per tape. All the memory technologies considered are based on 32nm technology node.

As shown in Figure 5.6, the density of 1bitDWM is similar to STT-MRAM and that of multibitDWM is higher than both SRAM and STT-MRAM. When we compare the leakage power, we can see that spin-based memory technologies can achieve significant reduction in the leakage power consumption compared to SRAM due to their non-volatility.

When we compare the access latency of different technologies, we can see that both the read and write latency of 1bitDWM is comparable to that of SRAM cache. Due to MTJ-based write, STT-MRAM has very high write latency. On the other hand, domain wall motion based write is highly efficient and enables us to improve the write latency significantly. When we consider multibitDWM, the access latency is variable and it depends on the the number of shifts required to access the required bit from the multibitDWM bit-cell. In the worst case, the shift+read latency of multibitDWM is 25.6 times higher than SRAM.

Next, when we consider the read energies, the spin-based memories can achieve significant benefits due to reduced bitline and wordline capacitances arising from improved density of these memories. Finally, the domain wall motion based write is highly energy efficient and this enables us to achieve significant reduction in the write energy compared to SRAM and STT-MRAM based caches. Among the different technologies, multibitDWM is found to achieve the lowest read and write energies. Also, the shift operations in multibitDWM is found to be highly energy-efficient and their overhead on read and write energy consumption is found to be negligible.

5.3.1 Impact of bits/tape of multibitDWM

Varying the number of bits per tape presents an interesting tradeoff between cache area and access latency of multibitDWM-based cache. Figure 5.7 presents the

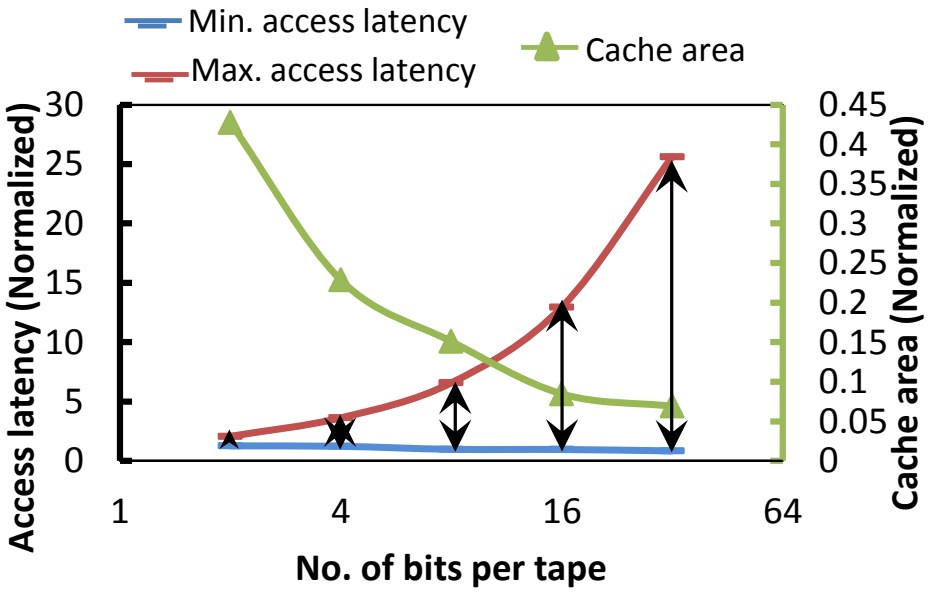


Fig. 5.7.: Impact of increasing bits/tape on cache area and access latency

impact of increasing the number of bits per tape on the access latency (shift + read) and area for a 128KB cache. As the number of bits per tape is increased, fewer multibitDWM bit-cells are required, leading to significant reduction in cache area. Further, the minimum access latency shows a small decrease due to the reduction in bitline capacitances with array size. However, the maximum access latency increases drastically due to the larger number of shift operations. This undesirable tradeoff between shift latency and density is unique to DWM and is a key challenge to the design of multibitDWM-based caches.

5.4 Conclusion

Spin-based memories have tremendous potential as future on-chip memories. However, the efficiency of write operation is a major bottleneck. In this chapter, we pro-

posed a new design– TAPESTRI – for designing spin-based cache that uses domain wall motion for performing writes. We proposed two different bit-cells—1bitDWM and multibitDWM—that are optimized for latency and density, respectively. We showed that 1bitDWM can outperform/match SRAM and STT-MRAM in all the key cache metrics. MultibitDWM, on the other hand, can achieve much higher density than all other bit-cell designs, but requires shifts before a read/write operation leading to variable access latencies. This is a major concern with multibitDWM and needs to be addressed through suitable circuit and architecture level optimizations.

6. TAPECACHE: CACHE DESIGN BASED ON DWM TAPES

In modern processors, a majority of the chip transistor count and area is occupied by cache memories. (*e.g*; 70% of the transistors in the Intel Core i3 are devoted to cache). The growing processor-memory gap along with increasingly complex applications and data sets, fuel an ever-increasing demand for larger caches. Caches also account for a significant portion of chip power consumption due to their significant leakage power. Consequently, spintronic memories that offer very high density and energy-efficiency are of great interest.

As described in the earlier chapters, DWM is a recently proposed spintronic memory technology that offers unprecedented density along with non-volatility. DWM was initially envisioned as a replacement for secondary storage due to its excellent density unmatched even among other emerging technologies. In this chapter, we make the first attempt to explore the use of domain wall memories as on-chip caches in general purpose computing platforms. Despite possessing a number of favorable features such as very high density, non-volatility, and low leakage, DWM has unique characteristics that pose significantly different challenges from all other memory technologies.

From an architectural perspective, a DWM device looks like a tape, which can store multiple (upto hundreds of) bits. Another key characteristic is that the bits stored in a DWM device/tape can be shifted in either direction. This enables the sharing of read/write ports across the bits stored in a tape, resulting in very high density. However, the time taken to access a bit stored in the tape depends on its location relative to the read/write port, leading to variable access latencies. For the bit stored at the read/write port, the access latency (best case) is lower compared to SRAM/STT-MRAM due to smaller bitlines resulting from higher density. However, the overall performance of a DWM cache is determined by the average number of shift

operations required per access. Hence, realizing a DWM-based cache requires the development of suitable circuit/architecture design techniques that exploit its strengths while reducing the performance penalties associated with shift operations. DWM, therefore, poses a fundamentally different challenge from other emerging memory technologies like STT-MRAM and PCRAM, which must be addressed.

The contributions of this work are as follows:

- We perform the first exploration of the design of on-chip caches using DWM. We propose TapeCache, a novel cache design in which all the levels in the cache hierarchy are realized using different DWM bit-cells (described in chapter 5). For latency-sensitive L1 cache, we design both the data and tag arrays using 1bitDWM that are optimized for latency. For L2 cache, we propose a hybrid organization in which the data array is designed using a combination of multibitDWM and 1bitDWM bit-cells, and the tag array is designed using 1bitDWM bit-cells.

- We propose circuit and architectural techniques to address the performance penalty arising from shift operations in multibitDWM bit-cells. Considering the read-write asymmetry and spatial locality of memory accesses found in most applications, we propose (i) a multi-port read-skewed multibitDWM bit-cell design that is optimized for reads, (ii) an efficient array organization and suitable cache management policies to maximally harness the performance potential of DWM-based caches.

- We perform an in-depth analysis of the power-performance tradeoffs present in TapeCache through architectural simulations of benchmarks with diverse read/write characteristics, miss rates and working sets. Our results show that an iso-capacity replacement of SRAM-based cache with TapeCache can result in large benefits in area and energy at iso-performance. We also demonstrate that DWM can significantly outperform STT-MRAM, which is considered to

be one of the most promising emerging memory technologies, in the context of on-chip cache design.

The rest of this chapter is organized as follows. Section 6.1 presents the proposed multi-port read-skewed multibitDWM bit-cell design. and discusses the corresponding the architectural implications. Section 6.2 describes the proposed cache architecture. Section 6.3 presents our methodology for modeling and evaluating TapeCache. Section 6.4 presents experimental results. and Section 6.5 concludes the chapter.

6.1 Multi-port read-skewed multibitDWM bit-cell

In this section, we present a multi-port read-skewed DWM multibitDWM bit-cell design enhanced with additional read-only ports that significantly reduces read access latencies, while retaining most of the density benefits associated with DWMs. It also helps expand the design space by introducing additional design parameters that can be utilized to perform more fine-grained energy-performance tradeoffs.

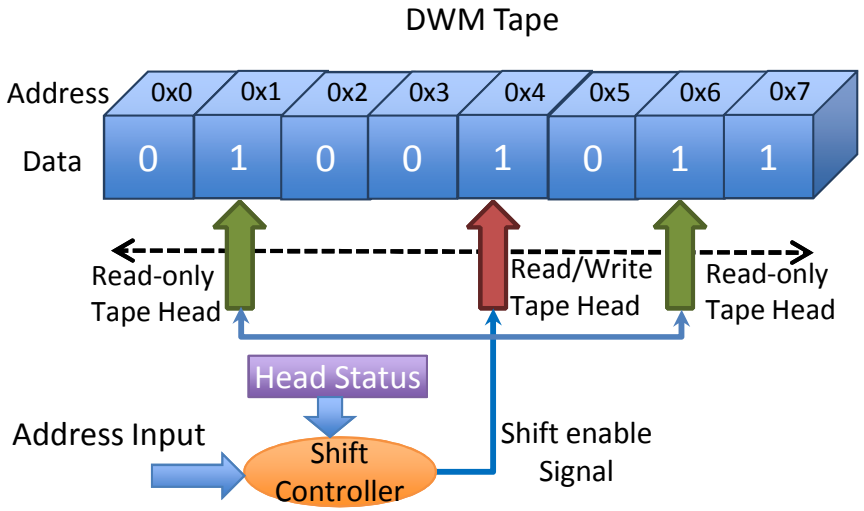


Fig. 6.1.: Logical view of a multi-port read-skewed multibitDWM bit-cell

The logical view of a multi-port read-skewed multibitDWM bit-cell is shown in Figure 6.1. It has multiple read-only tape heads distributed across the DWM tape.

In order to access a bit from the tape, the shift controller determines the appropriate tape head and computes the number of shift operations required by comparing the address bits with the current locations of the tape heads, referred to as the head status. It is important to note that there can be no relative movement between tape heads in a multi-port multibitDWM bit-cell as it is the bits stored in the tape that physically shift and not the tape heads. The multi-port configuration helps reduce the average number of shift operations per read access, thereby reducing the average read access time. The motivation for having additional read-only ports is two-fold: (i) from an architectural perspective, reads are more performance critical than writes and reads also outnumber writes (across a wide range of SPEC benchmarks store operations account for less than 25% of the total number of memory instructions). (ii) Read-only port can be realized using only two minimum-sized transistors, preserving the density benefits of DWM. For instance, adding a read-only port to a single-port multibitDWM bit-cell achieves 2X reduction in worst case read access latency with only 13% increase in area. On the other hand, halving the number of bits per tape achieves the same reduction in worst case read access latency with a 2X increase in area.

The proposed multi-port multibitDWM bit-cell design expands the design space and offers the following design parameters:

1. Number of bits per tape: Varying the number of bits per tape offers a tradeoff between density *vs.* worst case access latency for both reads and writes. Note that this tradeoff can be performed only at a coarse granularity as the number of bits per tape can be varied only in powers of 2 (to keep the addressing scheme simple).

2. Number of read-only tape heads: Varying the number of read-only tape heads alters only the read access latency (in contrast to bits per tape, which impacts read and write latencies), albeit in a more cost-effective manner. Furthermore, the number of read-only tape heads can be tuned in a more fine-grained manner as the number of read-only ports need not be in powers of 2.

3. Head Selection: The multi-port multibitDWM bit-cell provides the flexibility of selecting the best tape head for a given read operation, depending on the relative location of the tape heads and the bit to be accessed. Architectural policies for head selection are proposed in Section 6.2.3.

6.2 DWM cache architecture

The key decisions involved in the design of TapeCache include:

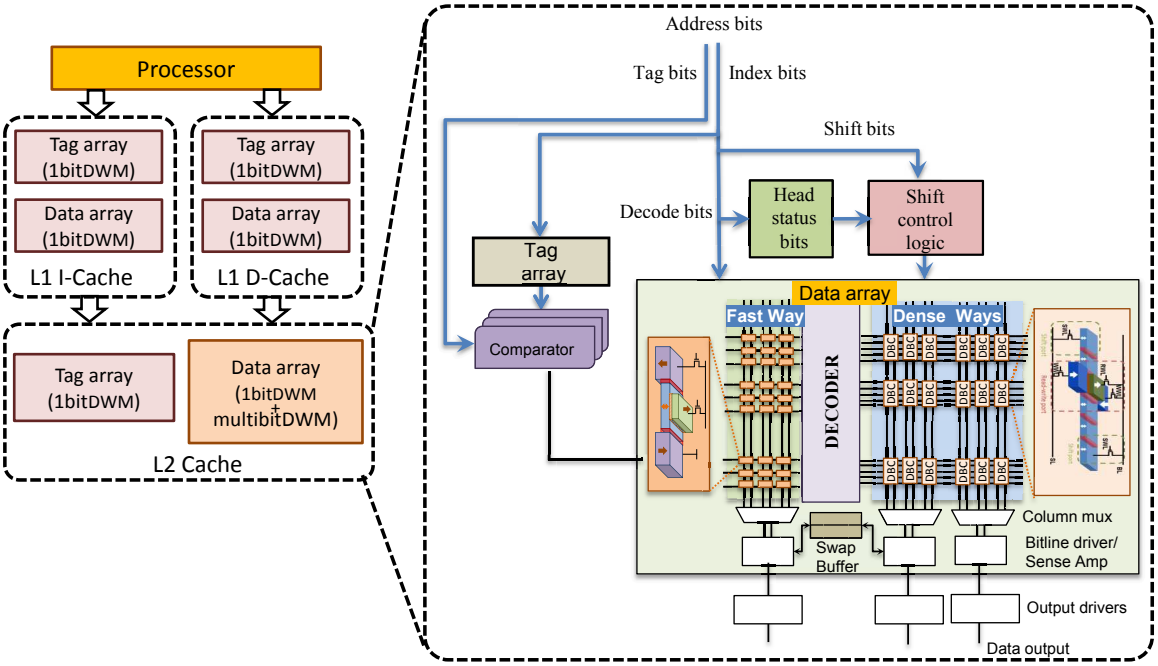


Fig. 6.2.: TapeCache organization

- **Choice of DWM cells:** 1bitDWM and multibitDWM bit-cells represent two different design points in the latency *vs.* density tradeoff space. One of these two design options must be chosen for each memory array in the cache hierarchy.
- **Data organization:** MultibitDWM is capable of storing multiple bits in a single bit-cell. Logically, a cache is divided into cache blocks, which must in turn be mapped to the multibitDWM bit-cells. This mapping scheme can store

bits from the same cache block in each bit-cell, or spread the bits of a cache block across multiple bit-cells.

- **Addressing policy:** Sharing of read/write ports across multiple bits in a multibitDWM bit-cell introduces the need for shifting bits before performing read/write operations. The addressing logic should not only select the multibit-DWM bit-cell to be accessed but also determine the number of shift operations required to access the required data.

- **Tape head management:** The additional latency caused by the varying number of shift operations for different bits stored in a multibitDWM bit-cell necessitates the use of suitable cache management policies to reduce the performance penalty.

Due to the above considerations, cache design with DWM differs significantly from traditional caches. In this section, we provide an overview of the proposed cache architecture and describe its key features.

Figure 6.2 depicts the overall organization of the proposed TapeCache architecture, which uses DWM to realize all levels in the cache hierarchy. Each level varies significantly in size and the required access latency. The L1 caches are responsible for providing fast access and its latency significantly impacts the overall performance of the system. Based on this consideration, we design both the data and tag arrays of the L1 caches (instruction cache and data cache) using latency-optimized 1bitDWM bit-cells, as shown in Figure 6.2. When we consider the L2 cache, it is responsible for reducing the number of off-chip accesses. The L2 cache is usually of much larger size, and its leakage contributes significantly to the total energy consumption. Considering the differing requirements, we propose a hybrid organization consisting of both 1bitDWM and multibitDWM bit-cells as described in Section 6.2.1. In addition, the L2 cache consists of a head status array, along with shift control logic that is used to manage the shift operations required to access data in each multibitDWM bit-cell. The multibitDWM bit-cells in the data array are grouped into DWM Block Clus-

ters (DBCs), which are capable of storing multiple cache blocks in a bit-interleaved fashion as explained in Section 6.2.2. In order to access a cache block, we need to select the correct DBC and determine the position of the block within that DBC. The traditional index bits are hence subdivided into decode bits and shift bits as shown in Figure 6.2. The decode bits are used to select the correct DBC and the shift bits are used along with the head status of the DBC to determine the number of shift operations required for accessing the cache block. In the following subsections, we present a detailed description of the L2 cache organization and management policies used in TapeCache.

6.2.1 Hybrid L2 cache design

Let us consider a simple L2 cache organization in which both the data array and the tag array are designed using multibitDWM. This design would result in maximum benefits in terms of both area and leakage power due to the high density and non-volatile nature of multibitDWM. However, the above configuration would require two variable latency operations per access, one for determining the block status from the tag array and the other for fetching the block from the data array[1]. This would considerably degrade the performance of the cache. Further, the area and energy benefits attainable from a multibitDWM-based tag array are relatively small, as the tag array represents a small fraction of the total area and power consumption of a cache. For instance, in a 1MB cache, the tag array contributes only 4.8% to the total cache area. Hence, we propose to design the tag array of L2 cache using 1bitDWM bit-cells.

In order to address the performance penalty from shift operations of multibitDWM bit-cells in the data array, we propose a hybrid organization in which we partition the cache into *fast ways* and *dense ways*. The fast ways of the cache are designed with 1bitDWM bit-cells, while the dense ways of the cache are designed with multibitDWM

[1]In lower level caches, the tag and data array access are serialized so as to avoid energy overheads associated with reading all ways of a cache simultaneously.

bit-cells, as shown in Figure 6.2. The motivation behind such an organization is to simultaneously exploit the latency benefits of 1bitDWM bit-cells and the density benefits of multibitDWM bit-cells. The fast ways are used to store frequently accessed cache blocks, thereby enabling lower latency access to performance critical data. The remaining cache blocks are stored in dense ways, resulting in an overall improvement in cache density.

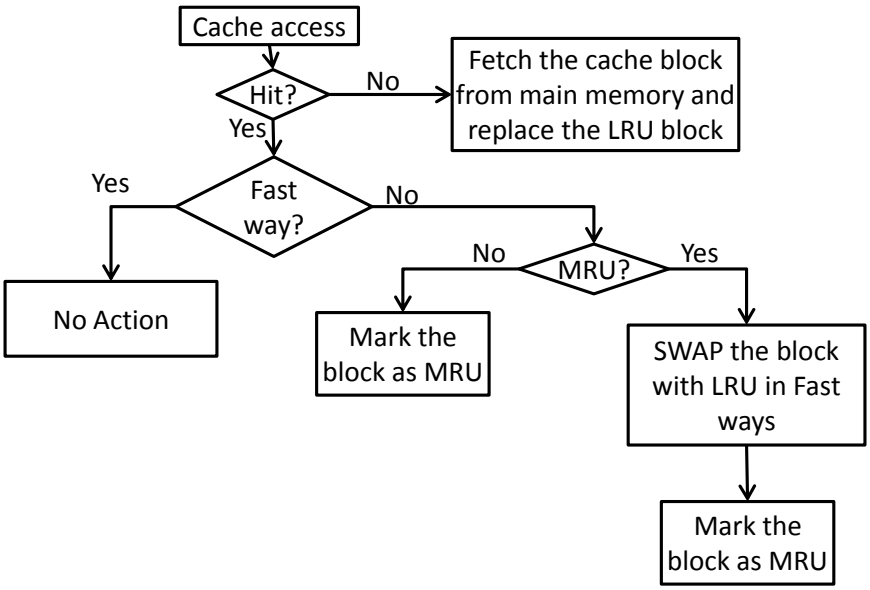


Fig. 6.3.: Hybrid L2 cache migration policy

Figure 6.3 demonstrates the working of this hybrid organization. When a cache line is hit in L2 cache, we check if the cache block is stored in a fast way or a dense way. If the cache block is present in a fast way, then no action is required. However, if the cache block is stored in a dense way, then it would require shift operations leading to higher access latencies. In this scenario, we need to determine if the cache block is frequently accessed and migrate it to a fast way if required. For this purpose, we check if the cache block has already been marked as most recently used (MRU) by the cache replacement policy. If so, this cache block is swapped with the LRU block in the fast ways. Therefore, in this cache migration policy, a cache block is determined to be frequently accessed only if a particular cache block is accessed consecutively twice

– the first access would update the cache block as MRU, and the second access would initiate the block swap to a fast way. A key benefit of this cache migration policy is that it utilizes the state information already maintained by the cache replacement policy. Therefore, the hardware overhead incurred by this scheme is negligible.

The swap operation between the fast and dense ways is implemented as follows. First, we determine if the cache block being accessed will initiate a swap operation based on the tag bits using the migration policy described above. If a swap operation is required, the LRU cache block from the fast ways ($Block_{LRU}$) and the accessed block ($Block_{acc}$) from the dense ways are read simultaneously and stored in swap buffers. After reading $Block_{acc}$ from the dense ways, we keep the tape heads aligned to the current location and write $Block_{LRU}$ immediately to this location, thereby eliminating the need for any additional shift operations. Simultaneously, $Block_{acc}$ is also written to the fast ways to complete the swap operation. In the proposed design, the swap buffer has only two entries and are implemented using 1bitDWM bit-cells, resulting in negligible hardware overhead. The performance penalty from the swap operations is also greatly reduced (only 2 cycles per swap) by (i) eliminating the need for additional shift operations, and (ii) overlapping the accesses to the fast ways with those of the dense ways. Also, the excellent energy efficiency of the proposed bit-cells greatly reduces the energy overhead associated with the swap operations. Therefore, the overheads from the proposed design are found to be negligible and are included in our evaluation in Section 6.4.

6.2.2 Bit-interleaved DWM Block-Cluster organization

Let us consider how the multiple bits in a cache block can be mapped to the bits in a DWM tape (multibitDWM bit-cell). One possible scheme involves mapping all the bits in a cache block to the same DWM tape (if cache block is larger than a tape, then it can be stored in multiple tapes). In this scenario, a cache access would involve N serial read/write operations, where N is the number of bits stored in a

DWM tape. This mapping would incur a very high access latency compared to a traditional SRAM-based cache.

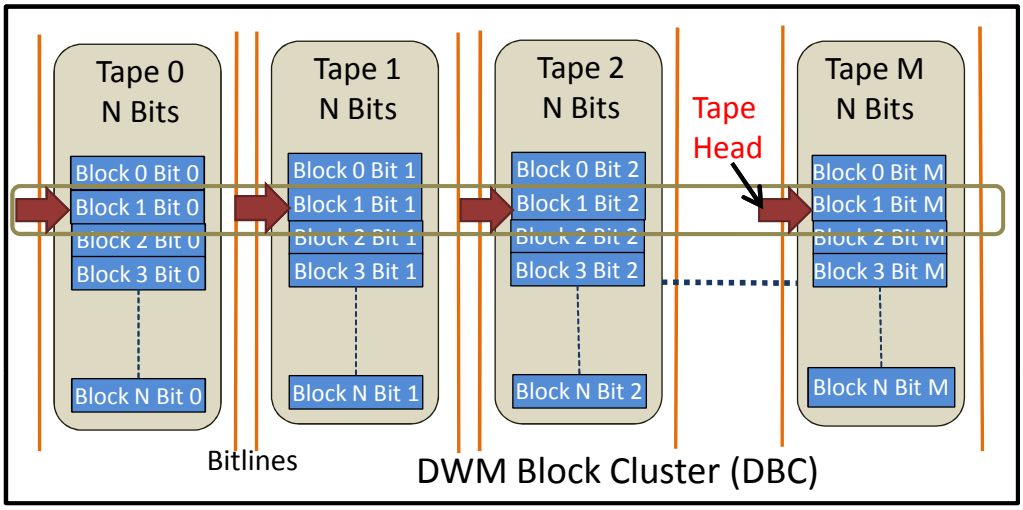


Fig. 6.4.: Bit-interleaved data array organization

In order to reduce the high latency overhead, we propose a bit-interleaved data array organization in which each cache block is spread across several DWM tapes. Figure 6.4 demonstrates how N blocks each of M-bits are stored in M DWM tapes each containing N bits of data. The i^{th} bit of each block is placed in tape number i. Within a tape, the j^{th} bit location stores a bit that belongs to block j. In this scheme, all the tape heads move in a lock-step fashion such that at a given time instance, all bits of some block are aligned to their respective tape's heads. Hence, in order to read a cache block, we perform the required number of shift operations and read all the bits belonging to the block in parallel. In this scenario, the number of shift operations required to access a block could vary from 0 to $N-1$, in contrast to the constant access latency of N observed in the previous scheme. The average access latency can be kept small in practice through the use of suitable head management policies, as discussed in the next sub-section. One of the overheads introduced by this scheme is the need for additional decoding logic to determine the number of shifts required to access a given block. However, our experimental evaluations indicate that

this overhead is negligible. Another overhead introduced by the bit-interleaved organization arises from performing shift operations in multiple tapes within a DBC for every cache access. Since the shift operation is highly energy-efficient [21], performing shift operations in multiple tapes result in negligible increase in the total energy consumption of the cache. Further, the tape heads in a DBC move in lock-step fashion, thereby amortizing the hardware overhead across multiple tapes in a DBC.

6.2.3 Head management policies

As described earlier, while accessing a cache block stored in a DBC, we need to perform an appropriate number of shift operations. The number of shift operations depends on the location of the block to be accessed relative to the position of the tape heads. Therefore, an effective head management policy is needed for reducing the performance overhead due to shift operations. In a multi-port multibitDWM cache design, head management policies involve (i) selection of the appropriate tape head for accessing the required data (tape head selection), and (ii) positioning of the tape head after the cache access (tape head update).

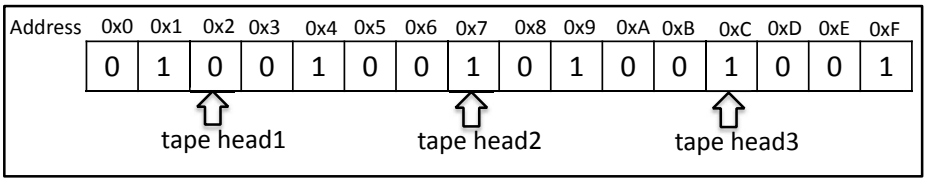
Tape head selection: In this work, we consider two different tape head selection policies – *Static* and *Dynamic*. In the *Static* tape head selection policy, each cache block is assigned a tape head statically depending on its initial location within the DBC. This policy has the advantage that the tape head can be determined solely based on the address of the cache block. On the other hand, in the *Dynamic* policy, we select the tape head that is nearest to the required cache block depending on the current DBC head status. For this purpose, we use the head status array to store the locations of tape heads and activate the appropriate tape head to access the data.

Tape head update: We explore three different tape head update policies: *Eager*, *Lazy*, and *Preshifting*. In the *Eager* policy, the tape heads are restored to their original default locations after each access. This policy simplifies the tape head selection, since we can assign the nearest tape head for every block statically. Also, this results in

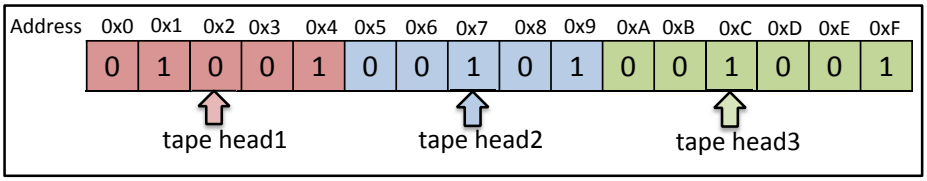
simplified shift control logic as the number of shift operations required to access a given cache block can be determined from the block address alone, and the head status does not need to be stored. In the *Lazy* policy, we do not restore the tape head to its default initial position after performing a read/write operation. Instead, we have status bits for each tape head to keep track of its location. When we perform a read/write operation, we calculate the difference between the block location within the tape and the current location of the tape head and then perform the required number of shifts. The *Lazy* policy is motivated by the spatial locality of memory accesses, which implies that the current tape head location tends to be closer to the next block to be accessed. In the *Preshifting* policy, we predict the next cache block that is likely to be accessed and align it with appropriate tape head. The concept of "preshifting" is similar to prefetching but is unique to DWM, which requires shifting operation for accessing the block.

The tape head selection and tape head update policies described above result in five distinct head management policies: *Static-Eager* (SE), *Dynamic-Lazy* (DL), *Static-Lazy* (SL), *Static-Preshifting* (SP), and Dynamic-Preshifting (DP). Note that *Dynamic-Eager* would be same as *Static-Eager* as the best port can be assigned statically for each location.

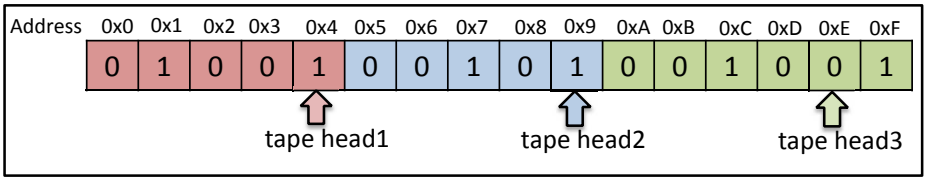
Figure 6.5 illustrates these five cache management policies using a DWM tape with 3 tape heads. For this illustration, we consider read operations, and therefore, the tape heads can be either read-only tape heads or read/write tape heads. Figure 6.5a shows the initial status. Figures 6.5(b-f) show the status of the DWM tape after performing a read operation at address 0x4. In Figures 6.5(b,c,e), the bits and tape heads are shaded to indicate the static head assignment. Let us now consider the tape head selection and number of shift operations required for the next access to address 0x5. The SE policy would use head2 and would require 2 left shift operations. The SL policy will also use head2 but requires 4 left shift operations. This shows that in the SL policy, a series of accesses to consecutive blocks would activate the worst case access latencies. The DL policy would choose head1 and require 1 right shift



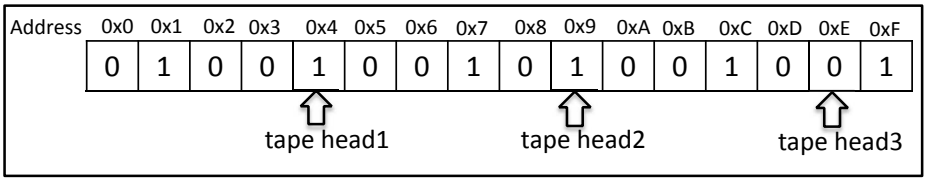
(a) Initial



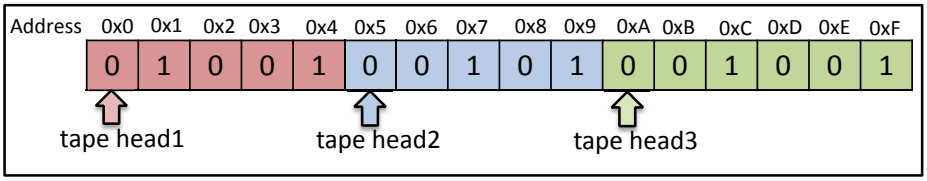
(b) After Read 0x4, Static-Eager



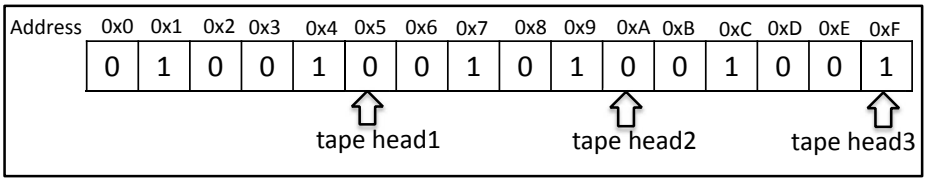
(c) After Read 0x4, Static-Lazy



(d) After Read 0x4, Dynamic-Lazy



(e) After Read 0x4, Static-Preshifting



(f) After Read 0x4, Dynamic-Preshifting

Fig. 6.5.: Comparison of different cache management policies

operation. Therefore, the DL policy exploits spatial locality, thereby reducing the average case shift latency overhead. However, this policy can skew the position of DWM tape head, thereby increasing the worst case latency for subsequent accesses.

This is overcome by restoring the DWM tape configuration during cache idle periods when it gets skewed beyond a certain threshold. For instance, access to address 0x0 after 0x4 would incur a large access latency with the DL policy. However, performing a restore operation after access to address 0x4 would shift tape head2 to address 0x4. This would enable the DL policy to exploit the spatial locality while eliminating higher access latencies due to skewed tape head position. In this work, we employ a low overhead scheme to determine the idle period of the cache by checking if the cache access queue is empty. The skew threshold for restoring the tape heads is determined empirically (In our experiments, we found 4 to be a suitable threshold). When we consider the SP and DP policies, both these policies do not require any shifts as the preshifting is successful. However, the two policies would use different tape heads (SP uses tape head2 and DP uses tape head1) for accessing the cache block. Note that, the DP policy can also skew the configuration of the tape and can lead to large worst case access latencies. However, preshifting offers a unique flexibility that is not present in other update policies. Preshifting involves two tape head selections - (i) first, when we perform the preshift operation, (ii) second, when the preshift is unsuccessful. Therefore, different strategies can be employed at these two steps, taking into consideration the performance criticality of different operations (read/write). This, as we explain below, helps us to reduce the performance impact from increased worst case latencies.

Adaptive preshifting policy: Figure 6.6 shows the different steps involved in the proposed adaptive preshifting policy. Initially, each cache block is assigned a tape head port statically based on its location. After a cache access, we predict the next cache block likely to be accessed and preshift the block to its *statically* assigned tape head. The use of statically assigned tape head for preshifting reduces the skew in the tape configuration, resulting in reduced worst case latency (Figure 6.5e vs. Figure 6.5f). If the prediction is successful, then no shift operation will be required, resulting in faster cache access. However, if the preshift fails and the cache access is a read operation, then we determine the nearest read port to the required cache block

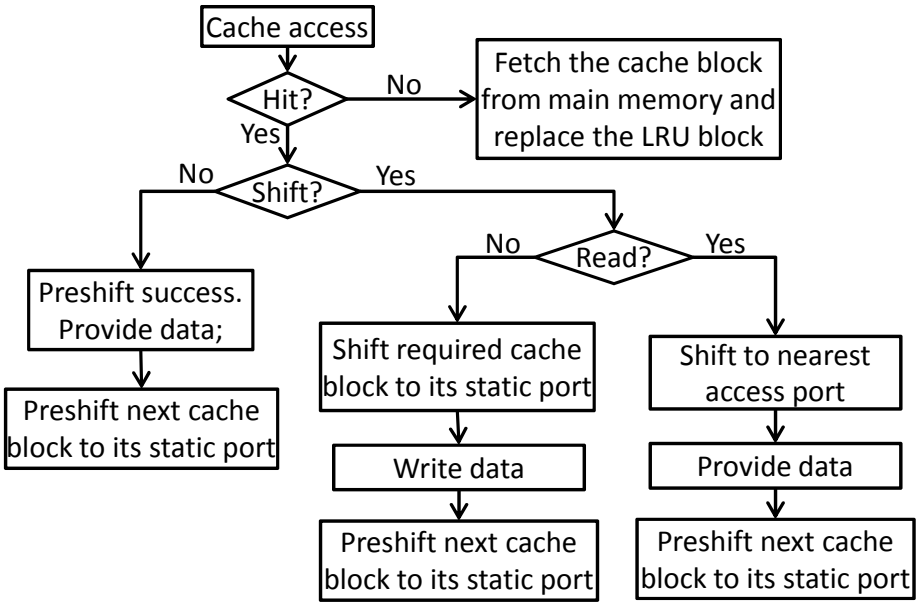


Fig. 6.6.: Adaptive preshifting policy

and use it to perform the required read access. This enables us to reduce the performance penalty due to misprediction for the performance critical read operation. On the other hand, if the cache access is a write operation, we use the statically allocated write port. This is because, it ensures that the number of extra bits required to avoid loss of data during shifting is small. Note that, a DWM tape storing N_b bits with N_{rw} read/write ports would require N_b/N_{rw} extra bits when we use statically allocated ports for writes compared to N_b extra bits if we use the nearest port for performing writes[2]. Also, a wide range of prediction mechanisms [125] can be employed to predict the next cache block for preshifting. In this work, we use a sequential prediction scheme. After performing an access to the cache block at address i, the location of the cache block that is likely to be accessed next is predicted as address $i+1$ and the corresponding tape heads are aligned to this location.

[2]Assuming that the number of shifts required for writes will be higher than that for reads due to the presence of read-only ports.

Note that head management policies are orthogonal to traditional cache management policies. For example, block replacement strategies such as LRU can be used unchanged in TapeCache.

6.3 Experimental methodology

In this section, we present a brief description of the modeling framework and then present the experimental setup used to evaluate TapeCache.

6.3.1 Modeling framework

TapeCache differs significantly from traditional memories in terms of both the device structure as well as cache architecture. In order to accurately evaluate the characteristics of the proposed cache design, we have developed a tool, Spin-CACTI, that is based on the CACTI framework [116]. First, we used a self-consistent device simulation framework [126] to accurately capture the domain wall motion using spin-torque and performed device level simulations of the DWM bit-cells. The DWM bit-cell parameters thus obtained were then used as technology parameters in Spin-CACTI to model the characteristics of TapeCache. The Spin-CACTI tool takes the number of bits per DWM tape, the number of read/write ports, and the number of read-only ports, along with the usual inputs to CACTI, to compute the area, energy and access latencies of TapeCache. Spin-CACTI takes into account the overheads due to the head status array, additional wordlines, and shift control logic while modeling TapeCache. The detailed description of the modeling framework is presented in Chapter 9.

6.3.2 Experimental setup

In our experiments, we perform an iso-capacity replacement for L1 and L2 caches and compare the area, energy and performance of the proposed design with that of

CMOS SRAM and STT-MRAM. All memory technologies considered are based on a 32nm technology node. The processor configuration used in our analysis is provided in Table 6.1. We evaluate SRAM memories using CACTI [116], STT-MRAM and DWM using our Spin-CACTI tool. We perform architectural simulations over a wide range of benchmarks from the SPEC 2006 suite using SimpleScalar [127] for 1 billion instructions after we warm up the cache by fast forwarding for 1 billion instructions.

Table 6.1.: System configuration

Processor Core	Alpha 21264 pipeline, Issue Width - 4
Processor Frequency	2 GHz
Functional Units	Integer - 8 ALUs, 4 Multipliers Floating Point - 2 ALUs, 2 Multipliers
L1 D/I-Cache	32KB, direct mapped, 32 byte line size, 2 cycle hit latency
L2 Unified Cache	1MB, 4-way associative, 64 byte line size, hit latency depends on technology

6.4 Experimental results

In this section, we present results comparing the benefits of using TapeCache with iso-capacity SRAM and STT-MRAM based caches. Note that DWM-based caches can also be used in an iso-area scenario wherein the density benefits can be translated into increased cache sizes for achieving higher performance. However, in this work, we focus on improving the energy and area efficiency of last level caches. We first present the results summarizing the benefits of proposed design in terms of area, energy and performance. We then present the results comparing the L1 and L2 cache characteristics of TapeCache with other memory technologies. Finally, we present architecture level results comparing the energy and performance of the proposed design across a wide range of benchmarks. In our experiments, we consider a DWM tape with 1 read/write port, 3 read-only ports, which is capable of storing 32 bits unless mentioned otherwise.

6.4.1 Results summary

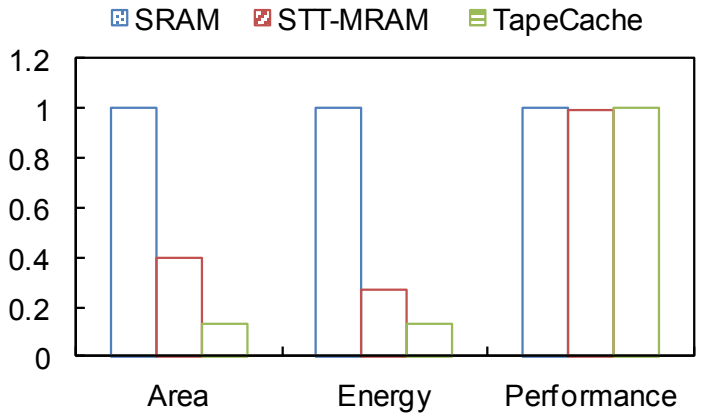


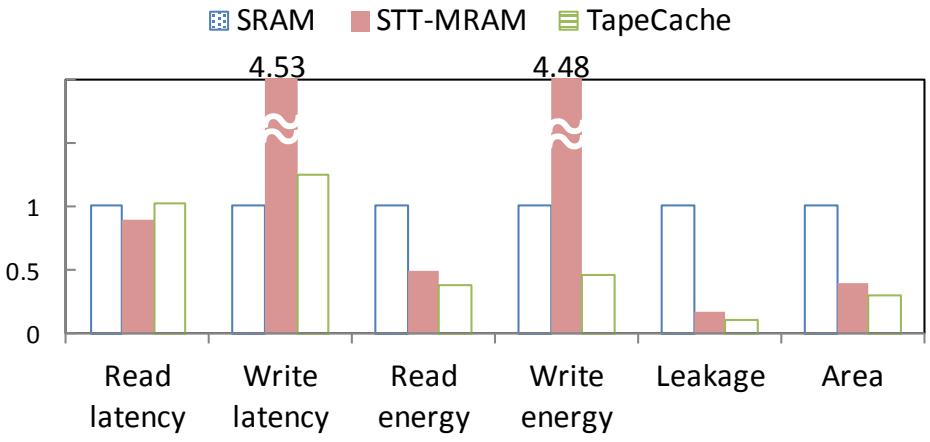
Fig. 6.7.: Comparison of area, energy and performance across different memory technologies

Figure 6.7 summarizes the benefits of TapeCache compared to SRAM and STT-MRAM based caches. Compared to SRAM-based cache, TapeCache achieves 7.5X improvement in energy and 7.8X improvement in area at virtually identical performance. When we compare the results with STT-MRAM based cache, TapeCache achieves 3.1X improvement in area and 2X improvement in energy along with a marginal performance improvement of 1.1%. Next, we will examine the benefits of TapeCache in greater detail.

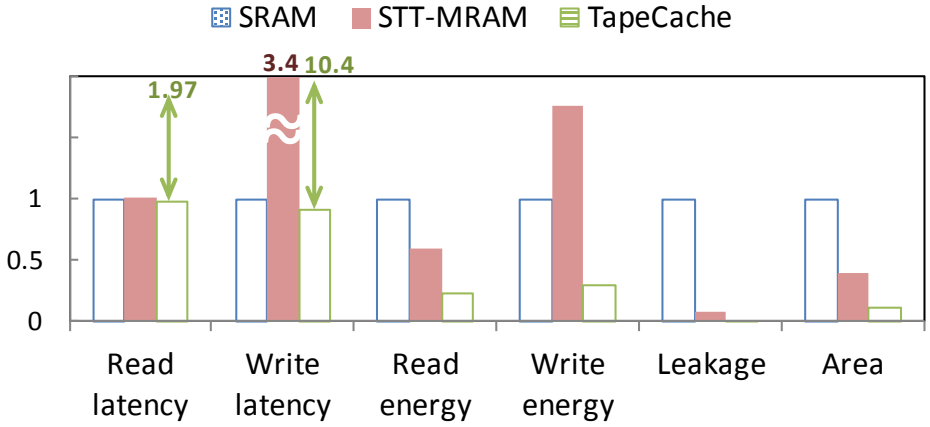
6.4.2 Cache characteristics

In this section, we present the results comparing the characteristics of the proposed L1 and L2 cache designs with that of SRAM and STT-MRAM based caches.

Figures 6.8a and 6.8b compare the L1 and L2 cache characteristics, respectively, across different memory technologies. As shown in the figure, the density of the L1 cache designed with 1bitDWM bit-cells is similar to STT-MRAM and that of the hybrid L2 cache designed with both 1bitDWM and multibitDWM bit-cells is significantly higher than both SRAM and STT-MRAM due to the higher density of



(a) L1 Cache



(b) L2 Cache

Fig. 6.8.: Comparison of L1 and L2 cache characteristics

multibitDWM bit-cells. When we compare the leakage power, we can see that spin-based memory technologies can achieve significant reduction in the leakage power consumption compared to SRAM due to their non-volatility. When we compare DWM with STT-MRAM, the reduction in leakage power consumption is due to smaller peripheral circuitry. The wordline and bitline drivers in the STT-MRAM cache need to be sized larger due to the increased capacitive load from the large access transistors. This marginally increases the leakage power consumption of the STT-MRAM cache compared to TapeCache.

When we compare the access latencies of different L1 caches, we can see that both the read and write latencies of TapeCache are comparable to SRAM cache. Due to

the inefficiency of MTJ-based writes, the STT-MRAM based L1 has very high write latency. On the other hand, shift-based write is highly efficient and enables us to improve the write latency significantly. When we consider the access latencies of different L2 caches, the access latency of the TapeCache varies due to the variable access latency of multibitDWM bit-cell, with the best case being comparable to SRAM. The effectiveness of preshifting implies that average access latencies are close to the best case.

Next, when we consider the read energies, all spin-based memories achieve significant benefits due to reduced bitline and wordline capacitances arising from improved density. Moreover, the shift-based write is highly energy efficient and this enables TapeCache to achieve significant reduction in write energy compared to SRAM and STT-MRAM based caches.

6.4.3 Architectural evaluation

In this section, we present the architecture level results comparing the energy and performance of TapeCache with SRAM and STT-MRAM caches across a wide range of benchmarks.

Energy consumption of TapeCache: Figure 6.9 compares the energy consumed by TapeCache with SRAM and STT-MRAM caches, normalized to the STT-MRAM cache. As we can see from the figure, TapeCache achieves significant reduction in the total cache energy consumption compared to both SRAM and STT-MRAM. STT-MRAM based cache reduces the leakage and read energy while increasing the write energy. TapeCache achieves reduction in all the three energy components- leakage, read and write. In addition, the proposed design achieves even higher reduction in leakage and read energy compared to STT-MRAM caches as shown earlier. As a result, TapeCache enables us to achieve maximum benefits in the total energy consumption of cache.

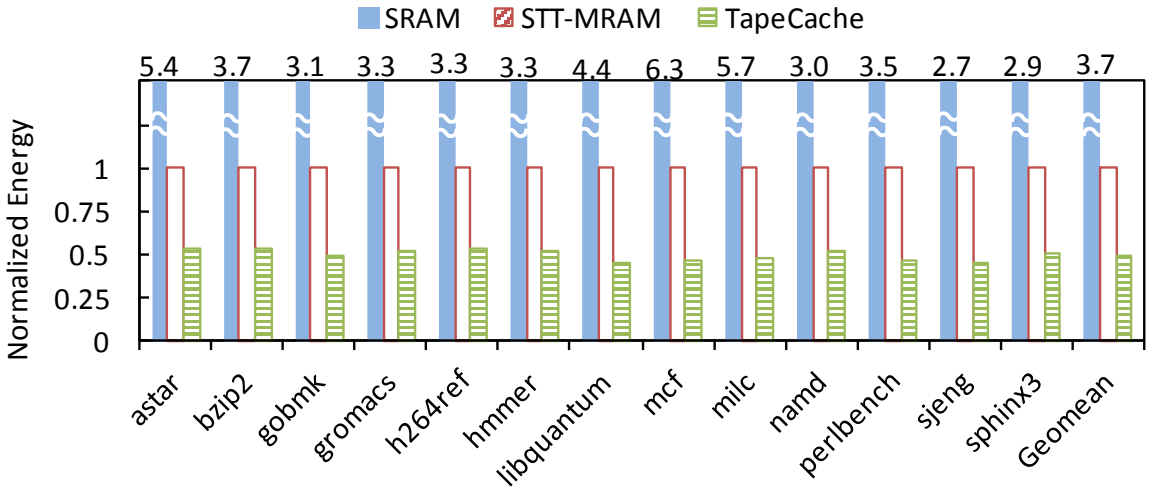


Fig. 6.9.: Comparison of energy consumption of cache across different memory technologies

TapeCache performance evaluation: One of the main challenges in designing L2 cache using multibitDWM bit-cells is the variable access latency due to shift operations. The performance of a TapeCache-based system depends on (i) the access latency to different bits in a DWM tape, and (ii) the access pattern, which determines the number of shift operations required. In the case of TapeCache, the improvement in density results in reduced access latency to the bits stored at the tape head compared to SRAM and STT-MRAM. This improves its best case access latency. The introduction of multiple ports reduces the number of shifts required to access the bits in a DWM tape, thereby reducing the worst case access latency. Further, the proposed cache management policies ensure that most of the cache accesses require smaller number of shifts, which reduces the average access latency of the proposed L2 cache. Figure 6.10 presents a comparison of the performance of different cache designs. Note that TapeCache results in performance improvement for benchmarks having high locality and predictable cache access pattern (gobmk, namd, perlbench, sjeng, sphinx3). On the other hand, for benchmarks (astar,libquantum, mcf, milc) that exhibit low degrees of locality, there is a reduction in performance due to increased shift penalty.

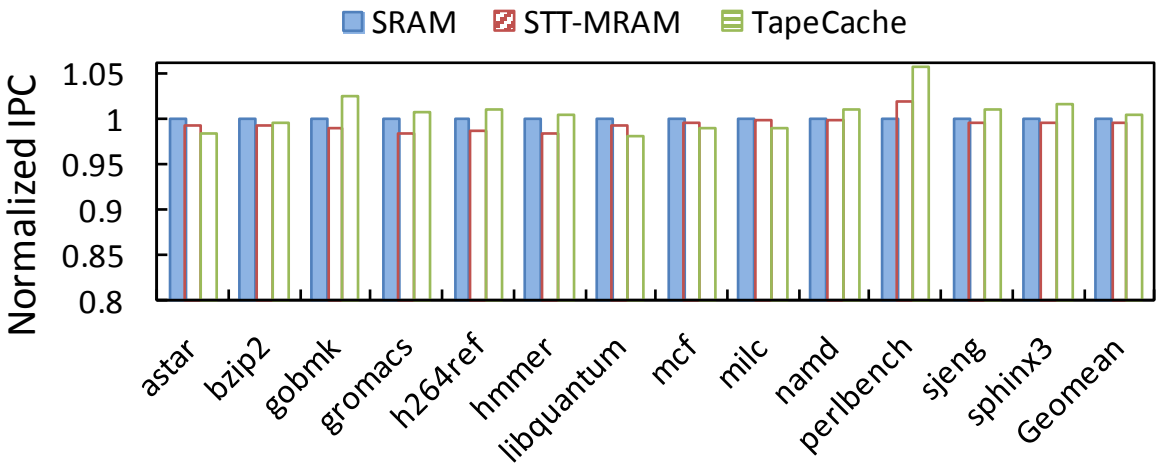


Fig. 6.10.: Performance comparison between different memory technologies

6.4.4 Design space exploration

Hybrid cache organizations: Using DWM, we can design two different hybrid caches: inter-layer and intra-layer. Inter-layer design uses different bit-cells to realize different levels in the cache hierarchy. Intra-layer design, on the other hand, uses different bit-cells even within a single level. In Section 6.2, we described an intra-layer hybrid cache organization. Inter-layer hybrid cache can be realized by designing the L1 cache with 1bitDWM bit-cells and the L2 cache with multibitDWM bit-cells. A comparison of these hybrid designs shows that the proposed intra-layer design achieves 23% improvement in performance over inter-layer design at comparable energy and area, thereby clearly underscoring the efficiency of the proposed intra-layer hybrid design.

Number of bits per tape: TapeCache offers a unique parameter –number of tape heads (read-only ports and read/write ports) – to tradeoff energy with performance. In Figure 6.11, we vary the number of read-only ports, read/write ports, and the tape head management policy and study their impact on cache energy consumption and performance. All the numbers in Figure 6.11 are normalized to that of SRAM cache. In the figure, a tape with x bits per tape, y read/write ports, and z read-only ports is denoted by x,y,z near different markers. As can be seen from Figure 6.11,

increasing the number of read-only ports and read/write ports reduces the average read latency while increasing the energy consumption of TapeCache. It is interesting to note the relationship between the average number of bits per read port and the tape head management policy. For configurations with smaller numbers of bits per tape head, the SE policy, which reduces the worst case access latency performs better than the DL policy. On the other hand, for configurations with higher numbers

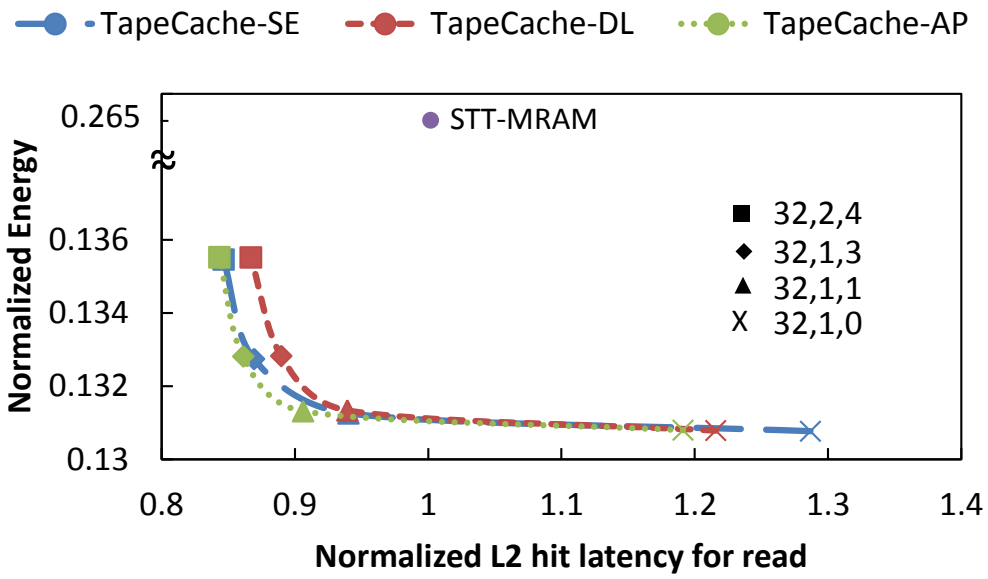


Fig. 6.11.: Design space exploration for TapeCache

of bits per tape head (less no. of read ports) the DL policy trumps SE since it exploits the spatial locality in access patterns. The proposed adaptive preshifting (AP) policy combines the benefits of both DL and SE policy and is found to be optimal across all the tape configurations. Also, note that the AP policy achieves significant improvement in performance ($> 9.5\%$) over the SE policy for (32,1,0) configuration, which has large number of bits per tape head (large variation in access latencies). As the number of tape heads per tape increases, the variation in access latencies reduces, resulting in diminishing benefits from the management policies. The figure also shows the corresponding iso-capacity design point for STT-MRAM based implementation, which is clearly sub-optimal to TapeCache.

6.5 Conclusion

Domain Wall Memory is an emerging spin-based memory technology that has a much higher density and good energy efficiency compared to current memory technologies (SRAM, DRAM), as well as other candidates for future memories such as STT-MRAM, PCRAM, *etc.* We explored the use of DWM for designing on-chip cache in computing platforms. The proposed cache organization and management policies reduce the performance penalty due to shift latency of DWMs. We performed architectural simulations to evaluate the benefits of a DWM-based cache. Our results demonstrate that DWM-based caches offer great potential in improving the energy-performance profile of a wide range of applications, while significantly reducing cache area.

7. STAG: SPINTRONIC-TAPE ARCHITECTURE FOR GPGPU CACHE HIERARCHIES

General Purpose Graphics Processing Units (GPGPUs) have emerged as efficient platforms to accelerate many highly parallel workloads, and are widely used across the spectrum of computing platforms, from mobile devices to supercomputers. With the growing interest in GPGPU computing, the number of cores in GPGPUs has exponentially increased over the years, leading to the doubling of theoretical peak performance with each generation. However, the ability of the memory sub-system to feed data to the cores has become a key determinant of system performance. The criticality of the memory sub-system is only expected to increase in the future, driven by growth in the number of cores on the one hand, and increases in the data sets processed by GPGPU applications on the other. The integration of GPUs with multi-core processors and into SoCs with shared memory systems further exacerbates the scarcity of off-chip memory bandwidth. To address this challenge, designers are driven to use increasing amounts of on-chip memory in GPUs. In addition to the increase in capacity, the on-chip memory has also evolved in complexity from a single-level software controlled scratchpad to a more sophisticated hierarchy with various hardware managed caches. In effect, a significant and increasing portion of GPGPU chip area and power budgets are being devoted on-chip memories.

In this chapter, we explore the use of DWMs to design the on-chip memory hierarchy of GPGPUs. We propose STAG, a new GPGPU cache architecture, which effectively leverages the intrinsic density and energy-efficiency of DWM devices and utilizes several architecture-level techniques to address their unique challenges. We demonstrate that these architectural optimizations enable STAG to outperform SRAM and STT-MRAM in both performance and energy across a wide range of GPGPU workloads.

We propose an all-spin cache architecture in which we design the latency-sensitive first level of the hierarchy, consisting of the L1 data cache, instruction cache, constant cache, and texture cache, using 1bitDWM, and design the second level using a hybrid organization consisting of a MultibitDWM-based data array and a 1bitDWM-based tag array. This hybrid organization largely harnesses the density benefits of DWM, while avoiding excessive performance penalty that results from shift operations during tag lookup. However, the shift penalty incurred during data array access still remains a challenge. In order to alleviate this overhead, we propose various architectural optimizations:

- *Clustered, bit-interleaved organization:* The bits corresponding to a cache block are distributed across a cluster of DWM tapes thereby allowing them to be accessed in parallel, while also sharing the control logic for shift operations across the tapes in a cluster.

- *Tape head management policy:* The position of the tape head *i.e.*, the bit in the tape to which the read/write port is aligned, is managed to reduce the expected shift latency for subsequent accesses.

- *Shift aware promotion buffer:* Accesses to the L2 cache from the different streaming multiprocessors (SMs) in the GPGPU are predicted based on intra-warp locality and locations that would incur a high shift penalty are promoted to a smaller buffer.

In summary, the key contributions of this work are as follows:

- We propose STAG, the first Domain Wall Memory (DWM) based cache hierarchy for GPGPUs. STAG employs two types of DWM bit-cells *viz.* 1bitDWM and multibitDWM to design all levels in the on-chip memory hierarchy of GPGPUs, taking into consideration their differing design requirements.

- We explore several cache organization and management policies for STAG by studying the unique challenges posed by DWM's shift operations in the con-

text of the GPGPU memory sub-system and the key characteristics of GPGPU workloads.

- We perform a detailed experimental evaluation of STAG and a design space exploration to analyze its sensitivity to different circuit and architectural parameters. We demonstrate that STAG achieves significant benefits in performance (12.1% on an average) and energy (3.3X on an average) over SRAM across a wide range of programs from the Rodinia, ISPASS and Parboil benchmark suites.

The rest of the chapter is organized as follows. Section 7.1 describes the STAG cache architecture and the various techniques employed to address the impact of shift operations. Section 7.2 explains the modeling framework and experimental methodology used to evaluate STAG. The results of our experiments are presented in Section 7.3 and Section 7.4 concludes the chapter.

7.1 STAG architecture

Current GPGPUs employ two levels in their on-chip cache hierarchy. The first level is comprised of a wide variety of caches such as instruction cache, data cache, texture cache and constant cache, which are exclusive to clusters of cores, called streaming multi-processors (SMs). These L1 caches represent a sizable portion of the total energy consumption of the on-chip memory hierarchy (55% in the our experiments). Moreover, their access latencies are critical to overall application performance. The next level in the hierarchy is formed by a unified L2 cache that is shared across all SMs. The L2 caches are typically large (*e.g.*, 1.5MB in Nvidia's GK110 and 1MB in AMD's Radeon 290X), and they are responsible for improving the effective memory bandwidth to the SMs by reducing the number of off-chip memory accesses. Typically, the latency of L2 cache accesses is hidden by switching between multiple warps that are active in the SMs. However, a very large L2 access latency will result in all active warps being exhausted, thereby degrading performance.

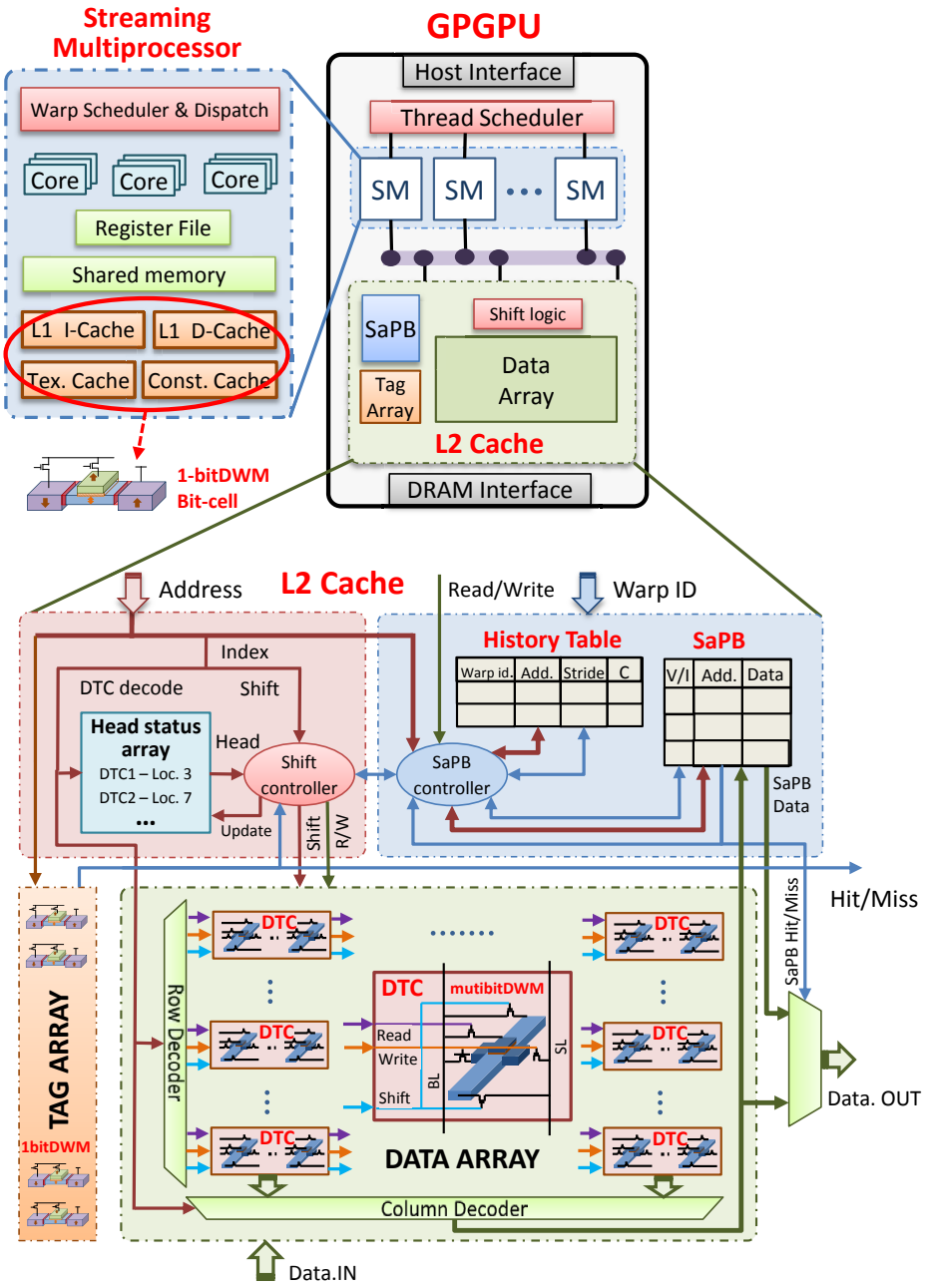


Fig. 7.1.: STAG architecture overview

We propose a high performance, energy efficient Spintronic Tape Architecture for GPGPU caches (STAG), shown in Figure 7.1, in which both levels of the cache hierarchy are designed using the DWM bit-cells presented in Chapter 5. The STAG

architecture, including organization and associated management policies, is described in the following subsections.

7.1.1 L1 cache design with 1bitDWM

Access latencies to the L1 caches greatly impact the performance of GPGPUs. Therefore, we utilize the 1bitDWM bit-cell to design the L1 caches in STAG. As described in Chapter 5, 1bitDWM outperforms SRAM in some characteristics (density, leakage power and dynamic energy consumption), while matching it in others (read/write latencies). A DWM cache designed to replace an SRAM cache may follow a spectrum of design alternatives, which range between two extrema: (i) an iso-cache-capacity replacement of SRAM arrays with 1bitDWM arrays, leading to improvements in cache area and energy, and (ii) an iso-area replacement of SRAM arrays with higher capacity 1bitDWM arrays, leading to reduction in miss rate and energy. While the later approach sounds appealing since lower L1 miss rates lead to improved performance and fewer L2 accesses, any increase in L1 hit latency severely degrades performance. Therefore, we choose a design point that lies in between these two extrema – increase L1 cache capacity as much as possible, while constraining access latency to be no more than the replaced SRAM cache. Following this strategy, we were able to use some, but not all, of the density improvement of 1bitDWM to increase L1 cache size (2X L1 capacity increase *vs.* a density increase of 3.6X).

The criticality of L1 access latency also precludes the use of multibitDWM bit-cells in the L1 caches. While this would vastly improve the density of L1 (allowing 10-30X larger L1 caches for the same area as SRAM), the latency overhead due to shift operations severely degrades application performance. Hence, as illustrated in Figure 7.1, STAG utilizes only 1bitDWM cells in the design of the first level of the cache hierarchy.

7.1.2 Hybrid L2 cache design

Density and energy efficiency are the most important considerations in the design of the L2 cache, making multibitDWM an attractive candidate. However, implementing the L2 cache exclusively using multibitDWM implies that accesses to the L2 cache would incur shift penalties in both tag array and data array accesses[1]. Our experiments suggested that this accumulation of shift penalties significantly degrades performance. Furthermore, in large L2 caches, the contribution of the tag array to the total cache area is small. Due to these factors, we propose a hybrid organization in which the tag array of the L2 cache is designed using 1bitDWM and the data array is designed using multibitDWM. Such an organization retains most of the density benefits offered by a complete multibitDWM design, while the worst case shift penalty is reduced in half. However, the shift penalty from the data array still remains a challenge and leads to increased L2 access latency, thereby impacting performance. We explore different architectural optimizations to address this issue in the rest of this section.

Bit-interleaved tape cluster organization

A straightforward realization of the L2 cache data array with multibitDWM would dedicate the requisite number of multibitDWM bit-cells to store the contents of each cache block. For example, a 64B cache block would be stored in 16 DWM tapes of 32 bits each. In this organization, a data array access would involve first selecting the group of tapes that store the block and then accessing the bits stored in each tape in a serial fashion. The hit latency of the cache would increase in direct proportion to the length of the DWM tapes (L). Our experiments show that this naïve DWM L2 cache organization leads to an average performance *degradation* of 6% compared to

[1]The data and tag arrays in lower level caches are typically accessed sequentially to save energy by reading only the required way rather than all ways.

an SRAM cache for $L = 32$, in spite of the lower miss rate due to the larger DWM L2 cache size. Using a lower L is undesirable, since it leads to lower density benefits.

To address the above challenge, we propose an alternative data organization, in which the bits of each cache block are mapped across a cluster of tapes in a bit-interleaved fashion, as shown in Figure 7.2. In this organization, a collection of DWM

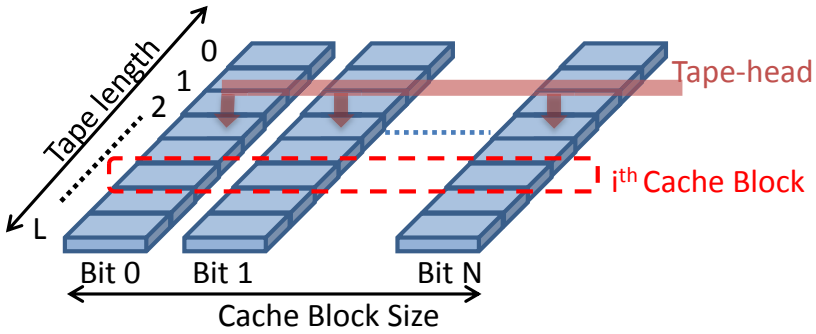


Fig. 7.2.: Bit-interleaved DWM tape cluster organization

tapes of equal length are grouped into a DWM tape cluster (DTC). The number of DWM tapes in a DTC is chosen to be equal to the number of bits in a cache block. Each tape in the DTC stores one bit of data from a given cache block in the same relative location within the tape *i.e.*, the i^{th} bits of all tapes in the DTC constitute a cache block. The number of cache blocks that can be stored in a DTC is equal to the length of the tape.

A key benefit offered by the proposed organization is that the bits of each cache block can be accessed in parallel from the DTC. A cache block is accessed by locating the DTC in which it is stored, computing its location in the DTC with respect to the tape heads, and shifting the tape heads of the DTC to the appropriate location, and reading/writing the block. As shown in Figure 7.1, the index bits in a cache block address are divided into two components *viz. decode bits* and *shift bits*. The decode bits are used to identify the correct DTC; in parallel, they are used to index a head status array, which stores the current location of the tape heads in each DTC. Note that the tape heads of all DWM tapes in a DTC move together in a lock-step fashion. Therefore, each DTC requires only a single entry in the head status array. The shift

bits, along with the head status of the relevant DTC, are used by the shift controller to determine the number of shifts required to access the cache block. The shift controller is shared across all DTCs in a cache bank. With the proposed organization, the time required to access a cache block in the L2 cache varies depending upon the number of shifts required. The average L2 cache access time is computed as follows:

$$AMAT_{L2} = T_{tag-1bitDWM} + missrate * T_{DRAM}$$
$$+hitrate * (T_{access-MultibitDWM} + (\sum_{K=1}^{L-1} p_K * K) * T_{shift}) \quad (7.1)$$

In the above equation, the first term ($T_{tag-1bitDWM}$) represents the tag lookup time, the second term represents the penalty due to L2 misses, and the third and fourth terms represent the latency for data array access during cache hits. $T_{access-MultibitDWM}$ represents the latency to read/write the cache block located at the tape heads and T_{shift} is the time taken to shift the tape by one position. ($\sum_{K=1}^{L-1} p_K * K$) represents the average number of shifts operations required, which depends on the probability (p_K) of a cache access requiring K shifts. This probability distribution directly depends on how the tape head is positioned relative the data to be accessed. *Thus, managing the location of the tape head plays a critical role in determining the average access latency of the L2 cache and hence the overall performance.*

Tape head management policies

Tape head management policies are designed to minimize the number of shift operations required to access the bits stored in a multibitDWM bit-cell, thereby reducing the overall access latency of the L2 cache. Towards this end, previous efforts that employ DWMs to design the cache hierarchy of scalar microprocessors [15, 16] have explored two tape-head management policies *viz.* restored head policy and leave-in-place head policy. We provide a brief description of these policies and argue that they lead to limited benefits in the context of massively parallel architectures such

as GPGPUs due to conflicting access requests from multiple SMs. To address their shortcomings, we then propose two key optimizations *viz.* preshifted head policy and shift-aware promotion buffer.

Restored head policy: In the restored head policy, as the name suggests, the tape heads of a DTC are restored back to the middle of the DWM tapes after each access to the DTC. This reduces the number of shifts required in the worst case to half of the tape length. Further, since the tape head is always at a fixed position before each access, there is no need to store the tape head status; the direction and the number of shifts required are determined solely from the shift bits in the block address.

Leave-in-place head policy: In the leave-in-place head policy, the tape-head is retained at the bit location that was most recently accessed in the DTC. This policy exploits the locality of data accesses to reduce the number of shift operations, as subsequent accesses to the DTC are more likely to be closer to the block that was recently accessed.

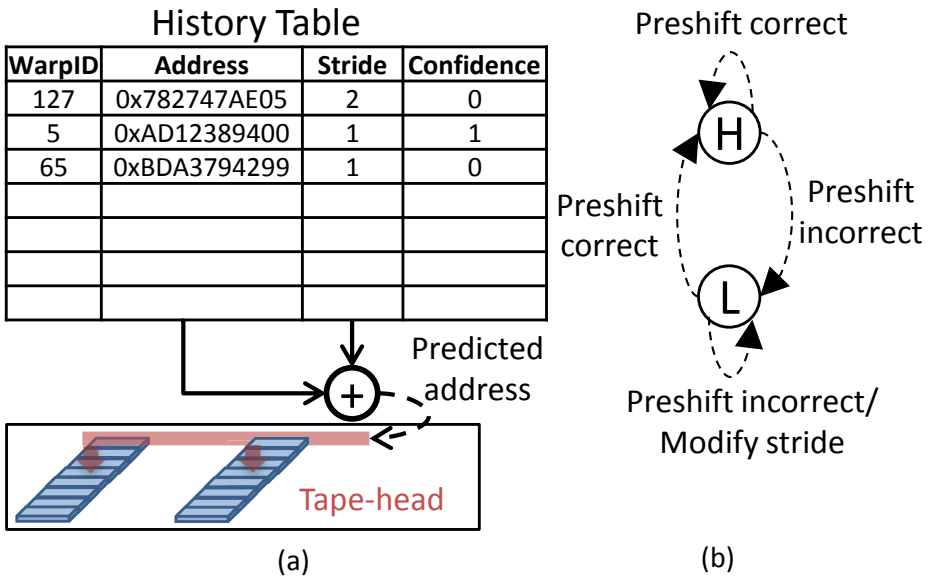


Fig. 7.3.: Preshifted head policy

Preshifted head policy: The above policies do not fully leverage the access characteristics of GPGPU applications. In GPGPUs, data accesses from a given warp

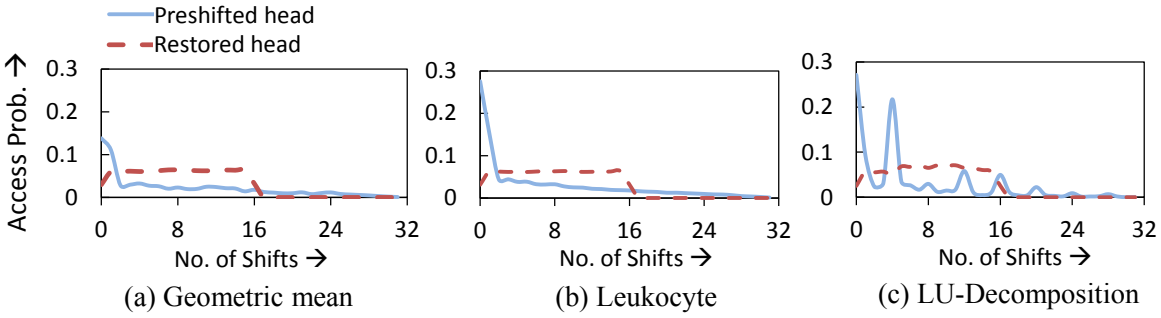


Fig. 7.4.: Probability distribution of shifts in different benchmarks

to the L2 cache typically exhibit high data locality [128]. However, since each SM executes multiple warps in a time-multiplexed fashion, the cache requests from different warps appear in an interleaved fashion. Since the above policies are oblivious to cache requests originating from multiple sources, they perform poorly in the context of GPGPUs. Therefore, we propose to enhance the above policies by predicting the data block that will be subsequently accessed in a DTC based on the warps that are currently active in a given SM and preshifting its tape head dynamically such that the shift penalty is minimized. Note that the prediction does not have to be completely accurate. We can benefit from preshifting even if the prediction results in the tape head moving closer to the block accessed next.

In STAG, we propose a low-overhead prediction scheme that utilizes the locality between cache accesses in a warp (intra-warp locality) for this purpose. In this scheme, each SM sends a message to the L2 cache with the warp ID of the currently active warp(s) well before the memory access from the SM reaches L2 cache. The L2 cache has a history table that contains one entry per active warp in the GPU. Each entry in the history table contains three fields: (i) the address of the most recent access from the warp, (ii) the address stride that is dynamically predicted for the warp and (iii) a single bit (low/high) that denotes the confidence in the predicted address stride. When a warp is first scheduled on an SM, an entry corresponding to its warp ID is allocated in the history table and the address stride is set to one with the confidence field set as low. When the first access arrives from the warp, it is stored in the address

field of the history table. The next access from the warp is predicted as the sum of the address in the history table and the predicted address stride. The tape head of the DTC that stores the predicted address is then preshifted to align with this address. If the next access from the warp matches the preshifted address, then the address stride is left unmodified and the confidence is set to high. However, if the prediction is unsuccessful and the confidence in the original prediction was low, the address stride is modified to the difference between the addresses of the current and the previous accesses. The confidence of the new prediction is set as low. Alternatively, if the confidence in the original prediction was high, then the address stride is left unmodified but its confidence level is demoted to low. In effect, the confidence field serves to filter out rare random accesses.

Figure 7.4 shows the resulting probability distribution of the number of shifts required over all L2 cache accesses, under the proposed tape head management policies. Figure 7.4(a) presents the average across all benchmarks, while Figures 7.4(b)-(c) present the results for two representative benchmarks. The DTCs used in the experiment contain 32 bits per tape with a cache block size of 128 bytes. As seen from the figure, when using the restored head policy, all accesses require a maximum of 16 shifts, since the tape head is maintained at the center of the tape. However, the number of shifts required is roughly uniformly distributed, resulting in a considerable number of shifts on average. On the other hand, the preshifted head policy, by leveraging the cache access patterns, is able to cater to a large fraction of accesses with very few (0 or 1) shift operations. However, a non-trivial number of cache accesses incur shift penalty that is closer to the worst case, which is equal to the length of the tape (twice that of restored head policy). This can be attributed to two factors. The first (intuitive) reason is that the prediction completely failed, moving the tape head in a direction opposite to the correct block. The second (non-intuitive) reason is that the L2 cache is shared between all SMs, with each running multiple warps in a time-multiplexed fashion. When the operating sets of different warps reside on the same DTC, even if the prediction is correct for a given warp, the DTC may incur

a large shift penalty because the next cache request is from a different warp or SM. This phenomenon is demonstrated in Figure 7.5 for the neural network application from the ISPASS benchmark suite. Figure 7.5 shows the cache blocks accessed in one of the DTCs over time. The accesses are color coded based on warp ID *i.e.*, all accesses of the same color arrive from the same warp. The patterns indicate that the application has good intra-warp locality and one would expect the preshifted head policy to perform well. However, accesses from several warps are intermingled (as shown in the inset) and as a result, the DTC incurs large shift penalties. In order to account for these cases, we propose a *Shift aware Promotion Buffer (SaPB)*, which simultaneously exploits intra-warp locality while reducing the worst case access latency.

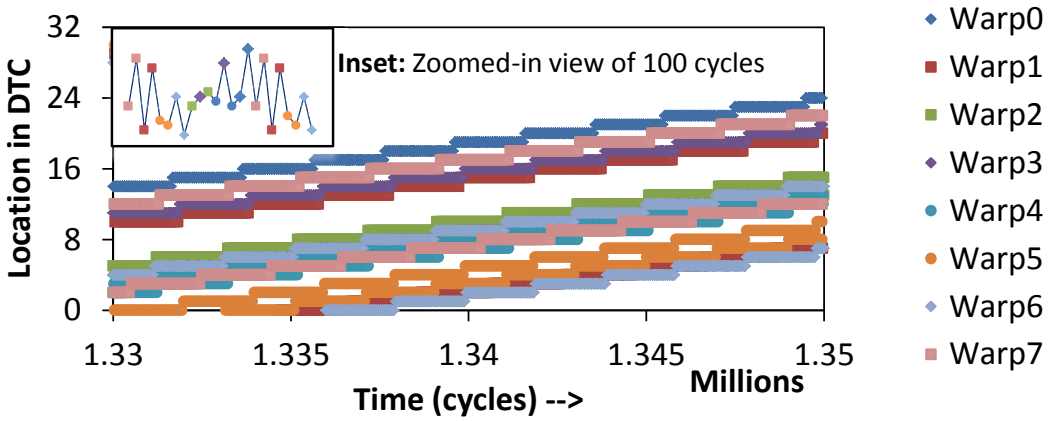


Fig. 7.5.: Access pattern of neural network benchmark

7.1.3 Shift aware promotion buffer

Shift aware Promotion Buffer (SaPB) is a small, fully-associative buffer that is introduced in the cache hierarchy before the L2 cache to reduce the number of long latency accesses. Similar to the L2 cache, the SaPB is shared by all the SMs in the GPU. The SaPB is designed using 1bitDWM and is hence fast and energy-efficient. The basic tenet behind SaPB is to selectively promote cache blocks from the L2 cache

that incur large number of shifts to a smaller buffer so that they can be accessed with lower latency.

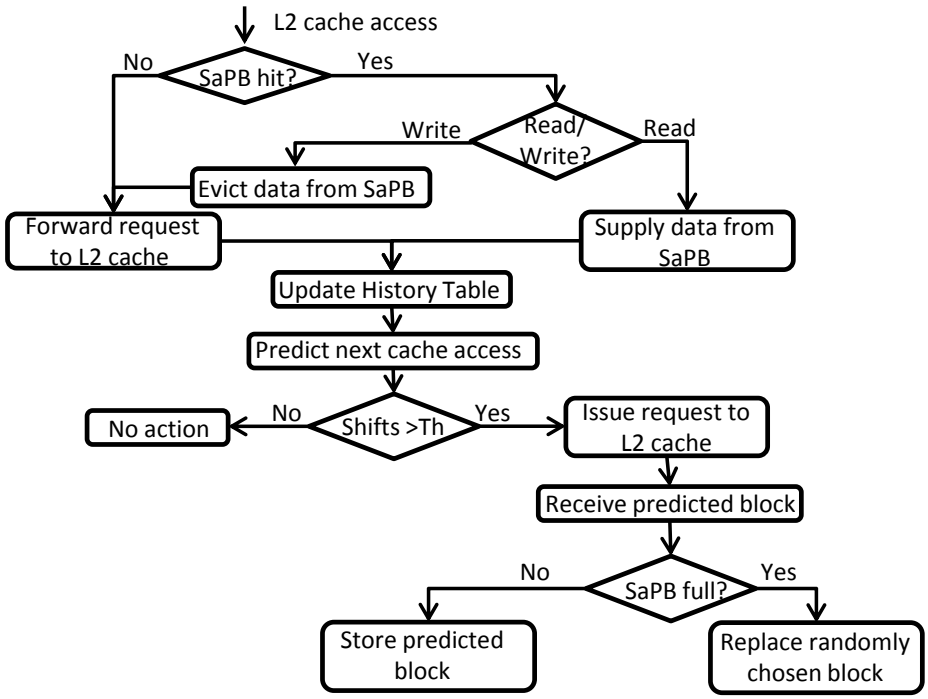


Fig. 7.6.: SaPB operation

Figure 7.6 outlines the steps involved in the operation of the SaPB. Every access to the L2 cache is first checked in the SaPB for a hit/miss. If the access is a hit and it is a read access, then the data is directly provided from the SaPB. If a write operation is a hit in the SaPB, the block is evicted and the write proceeds as normal to the L2 cache. This write evict policy is adopted to ensure that none of the blocks in the SaPB are dirty. If the access is a miss in the SaPB, then the request is forwarded to the L2 cache. We note that for every access to the L2/SaPB, the history table entry corresponding to the warp that issued the request is updated and the next predicted block is promoted to the SaPB. If the SaPB is full, a random block is chosen to be replaced.

SaPB is a unique design choice for DWM-based caches because, unlike SRAM or STT-MRAM caches, they incur variable latencies based on where the block is

stored relative to the tape head. In order to effectively utilize the SaPB, we need to promote only the blocks that (i) are likely to be accessed in near future, and (ii) would incur significant shift penalties in the L2 cache. In STAG, we utilize the warp ID based prediction scheme proposed in the preshifted head policy to determine suitable candidates for promotion. We then determine the number of shifts required for their access and only promote those requiring a number of shifts greater than a threshold. In our experiments, we found 4 to be a suitable choice for the promotion threshold. Since likely-accessed blocks are already promoted to the SaPB, we employ the restored head policy in the DTCs, thereby ensuring that the worst case latency is reduced to half the tape length even during mis-predicts. The average access time of the hybrid L2 cache with the SaPB and restored tape head management policy in the DTCs is given in Equation 7.2.

$$\begin{aligned} AMAT_{SaPB} &= T_{SaPB-tag} + hitrate_{SaPB} * T_{SaPB-data} \\ &+ missrate_{SaPB} * (T_{L2-tag-1bitDWM} + missrate_{L2} * T_{DRAM} \\ &+ hitrate_{L2} * (T_{access-MultibitDWM} + (\sum_{K=1}^{L/2} p_K * K) * T_{shift})) \end{aligned} \tag{7.2}$$

We note that the SaPB is quite different from enhancing the L1 caches with the ability to prefetch cache blocks. This is because: (i) The SaPB is "shift-aware" *i.e.,* it selectively provides faster access to only those cache blocks that require large number of shifts in the L2 cache. We do not store the cache blocks that require small number of shifts in the SaPB, as their latency can be effectively hidden in GPGPUs (by exploiting concurrency through warp-switching). Conventional prefetching would be agnostic to the number of shifts required and hence could promote/prefetch blocks that have very small or no impact on performance. (ii) L1 caches are private to a given SM, while the SaPB is shared among all SMs. Sharing enables better utilization of the SaPB across cache blocks from different warps. Also, prefetching would require increasing the capacity of L1 caches to avoid conflict misses, which adversely impacts

their access latency as they are already sized to maximum possible for single-cycle access.

In summary, the proposed architectural optimizations and cache management policies allow us to exploit the inherent benefits of domain wall memories and effectively translate them into energy and performance improvements in GPGPUs.

7.2 Experimental methodology

In order to evaluate STAG, we developed a device-to-architecture modeling framework that projects DWM device characteristics to the system level. In this section, we describe the framework and present the experimental setup used in our evaluation.

Table 7.1.: GPGPU configuration

No. of SMs	16
SM configuration	48 warps, 32 threads, 1.4 GHz,32768 registers, Scheduling: Round-Robin, Shared Memory: 48KB
L1 caches	Data cache: 16KB, 4 way, 128B block Instruction cache: 4KB, 4 way, 64B block Texture cache: 16KB, 16 way, 64B block Constant cache: 8KB, 2 way, 64B block
L2 cache	128KB/bank, 6 banks, 16 way, 128B block
Memory	6 memory controllers, FR-FCFS scheduling, 16 banks,Burst length=16
GDDR3 Timing	t_{CL}=12,t_{RP}=12,t_{RC}=40,t_{RAS}=28,t_{CCD}=2,t_{WL}=4, t_{RCD}=12,t_{RRD}=6, t_{CDLR}=5,t_{WR}=12,t_{CCDL}=3,t_{RTPL}=2

Table 7.2.: GPGPU workloads

Cache Capacity Sensitive (CCS)	Back propagation (bp), Breadth First Search (bfs), Needleman-Wunsch (nw), CFD Euler3D Double (cfded), Kmeans Clustering (kmeans), CFD Euler3D (cfde), CFD Pre-euler3D Double (cfdpd), Histogram (histo), Magnetic Resonance Imaging-Gridding (mrig), 3D Stencil Operation (stencil)
Cache Capacity Insensitive (CCI)	Heart Wall (hw), HotSpot (hs), Particle Filter (pf), Neural Network (nn), Path Finder (pathf), Leukocyte (lc), LU Decomposition (lud), Distance-Cutoff Coulombic Potential (cutcp), Magnetic Resonance Imaging-Q (mri-q), Two Point Angular Correlation Function (tpacf)

DWM cache modeling: For evaluating a standalone DWM cache, we used Spin-CACTI, a modified version of the popular CACTI tool [116]. Spin-CACTI takes as input various device characteristics, which are obtained using a physics-based device simulation model [126] that has been validated with measured experimental data. It computes cache characteristics including read, write, and shift latency/energy, area, and leakage power. Spin-CACTI captures several unique characteristics of STAG such as (i) hybrid data and tag array organization in the L2 cache, (ii) modified decoding logic in which row and column decoders are used to select a DTC instead of directly choosing a cache block in the case of traditional caches, *etc.* We used Spin-CACTI to evaluate different Spin-based L1 caches and the hybrid L2 cache.

For our baseline, we considered cache designs based on SRAM and STT-MRAM memory technologies. We evaluated the SRAM cache using the standard CACTI [116] and used Spin-CACTI for the STT-MRAM cache. In addition to modeling the on-chip memories, we also considered the impact of main memory on performance and energy and modeled its characteristics using CACTI. All our evaluations of DWM, SRAM and STT-MRAM memory technologies were performed at the 32nm technology node.

Hardware overheads: The history table, SaPB, and head status array were designed using 1bitDWM and Spin-CACTI was used for modeling their overheads. The hardware complexity of the shift controller depends on the tape-head management policy employed in the cache. For the restored head policy, the number and direction of shifts can be determined by simple bit-wise operations on the cache block address. For the preshifted head policy, a $log(N)$ bit wide subtractor (where N is the number of bits/tape) is required to compute the number of shifts. However, since the shift controller is shared by all DTCs in a cache bank, its overhead was found to be $< 0.1\%$ of the total cache energy.

Architectural simulation: In order to evaluate the impact of STAG at the application level, we used GPGPU-Sim v3.2.0 [129], a cycle accurate GPU simulator, evaluate a wide range of GPGPU workloads. The baseline GPU configuration considered in this work is shown in Table 7.1. In order to estimate the total energy

consumed, we used GPGPU-Sim to obtain L1 cache, L2 cache and main memory access traces for each benchmark. These traces were used with the memory characteristics obtained from CACTI/Spin-CACTI to evaluate the total memory system energy consumption. We considered an SaPB of size 64KB, and a history table with 256 entries in our design. The size of the head status array varied depending on the cache configuration. For an L2 cache designed with 32 bits per DWM tape, and 4MB per bank, the size of head status array was approximately 5KB.

Workloads: We used a wide range of GPGPU workloads (shown in Table 7.2) from the Rodinia [130], ISPASS [129] and Parboil [131] benchmark suites. The benchmarks in Table 7.2 are classified into cache capacity sensitive (CCS) and cache capacity insensitive (CCI) based on the impact of increased cache capacity. We performed all our simulations for a maximum of 2 billion instructions, or until program termination.

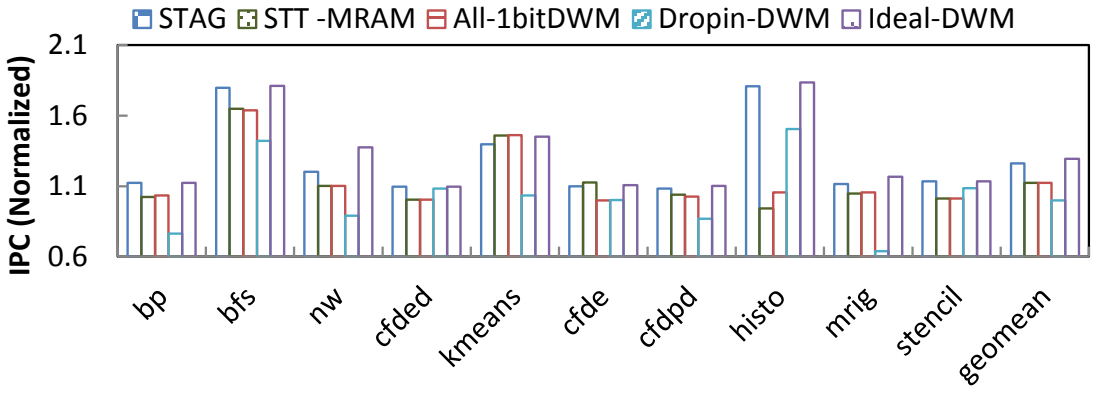
7.3 Experimental results

In this section, we present the results of various experiments that demonstrate the benefits of STAG. The performance and energy results are normalized to the baseline SRAM cache design, unless stated otherwise.

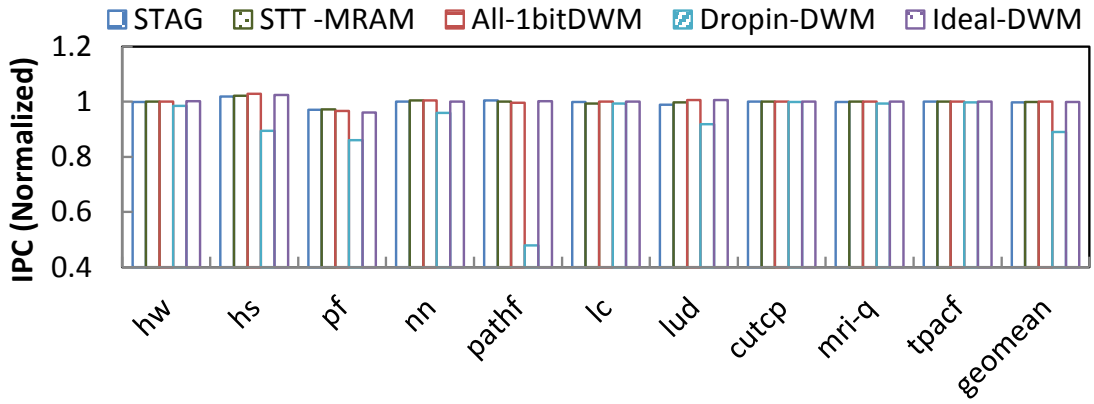
7.3.1 Performance comparison

Figure 7.7 shows the performance improvement (increase in IPC) obtained using STAG under iso-area conditions. For cache capacity sensitive (CCS) benchmarks, STAG achieves 26% improvement in IPC over SRAM. This increase in performance stems from three factors: (i) increased L1 cache capacity of 2X enabled by 1bitDWM, (ii) increased L2 capacity of 32X enabled by multibitDWM, and (iii) the proposed architectural optimizations that mitigate the performance penalty from shift operations in the L2 cache.

Figure 7.7 also compares the performance of STAG with four other baselines, *viz.* an STT-MRAM cache, a DWM cache where all levels are implemented using 1bit-



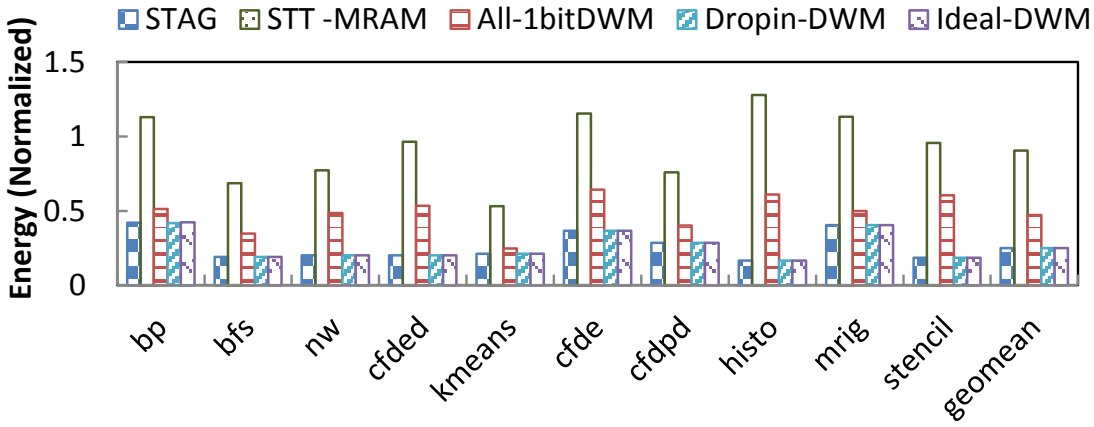
(a) Capacity sensitive



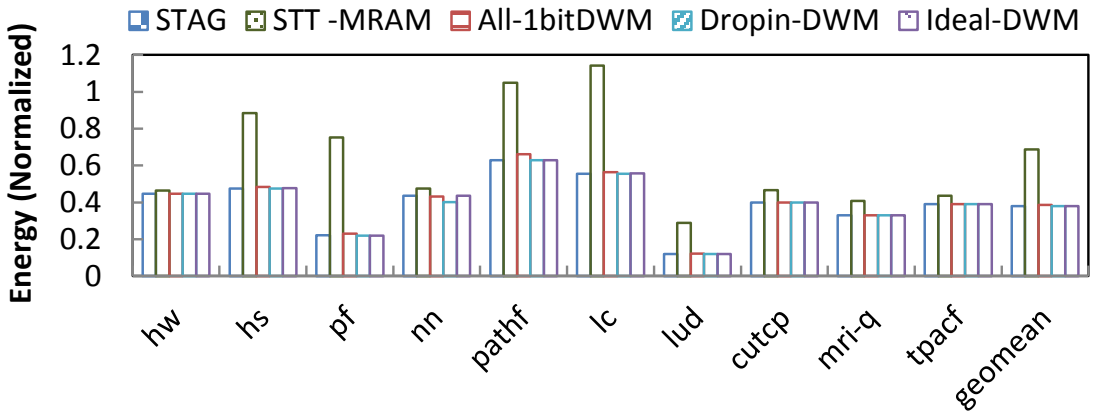
(b) Capacity insensitive

Fig. 7.7.: Performance of different cache designs under iso-area conditions

DWM bit-cells ($All-1bitDWM$), a DWM cache in which L2 cache blocks are directly replaced with multibitDWM cells without the proposed architectural optimizations ($Dropin-DWM$) and finally an ideal DWM cache ($Ideal-DWM$) that assumes zero penalties due to shift operations. Each of the caches are individually re-sized for iso-area. We observe from Figure 7.7 that STAG achieves 12.2% IPC improvement over the STT-MRAM and All-1bitDWM designs. Note that the density of STT-MRAM and 1bitDWM bit-cells are comparable and hence their impact on IPC are also very similar. Also, when multibitDWM cells are directly introduced at the L2 level, performance drops by 6% despite the increased L2 capacity. This underscores the need for the proposed cache organization and management policies, which alleviate the shift penalties. In the case of CCI benchmarks, all the cache designs were



(a) Capacity sensitive



(b) Capacity insensitive

Fig. 7.8.: Energy consumption of different cache designs under iso-area conditions

found to have similar performance, except for a few cases where the Dropin-DWM design displays poor performance due to excessive shift operations. Across both CCS and CCI benchmarks, STAG achieves 12.1% and 5.8% performance improvement over SRAM and STT-MRAM/All-1bitDWM, respectively, and reaches within 1.5% of the Ideal-DWM design.

7.3.2 Energy comparison

We now compare the energy consumed by STAG with the baselines described in Section 7.3.1. As shown in Figure 7.8, STAG achieves energy reduction of 4X

for CCS benchmarks and 2.63X for CCI benchmarks over SRAM. This is primarily attributed to the reduction in leakage and lower read/write energy. In addition, for CCS benchmarks, the higher capacity of STAG reduces the number of off-chip accesses to main memory by 9X leading to further energy reduction.

Compared to the STT-MRAM design, the energy benefits amount to 3.6X for CCS benchmarks and 1.8X for CCI benchmarks respectively. STT-MRAM, being a spin-based memory, consumes negligible leakage power. However, it consumes high energy during cache writes, which the shift-based write of DWM helps alleviate. Additional energy benefits of STAG in the case of CCS benchmarks are again attributed to the reduction in off-chip accesses. The All-1bitDWM design, owing to the optimized write energy of DWMs, is more energy-efficient compared to STT-MRAM.

In summary, the proposed cache design is able to exploit the strengths of DWM and simultaneously derive benefits in both performance and energy across a wide range of benchmarks.

7.3.3 Design space exploration

In this section, we perform a detailed design space exploration and analyze the sensitivity of STAG to different circuit and architectural parameters such as: (i) Bits per tape, (ii) tape-head management policy, and (iii) cache size.

Bits per tape

The number of bits stored in the multibitDWM tape is key to its density and performance. Therefore, we study the impact of varying this parameter at both iso-area and iso-capacity conditions. In these experiments, DWM-N denotes a design where N bits are stored in a DWM tape.

Iso-area design: Increasing the number of bits per tape under iso-area conditions enables us to design a cache of higher capacity at the cost of increased shift penalties. Therefore, the bits/tape parameter can have a pronounced impact on the IPC of

applications that are sensitive to either cache capacity or latency. Our exploration on CCS benchmarks, shown in Figure 7.9, reveals four major trends.

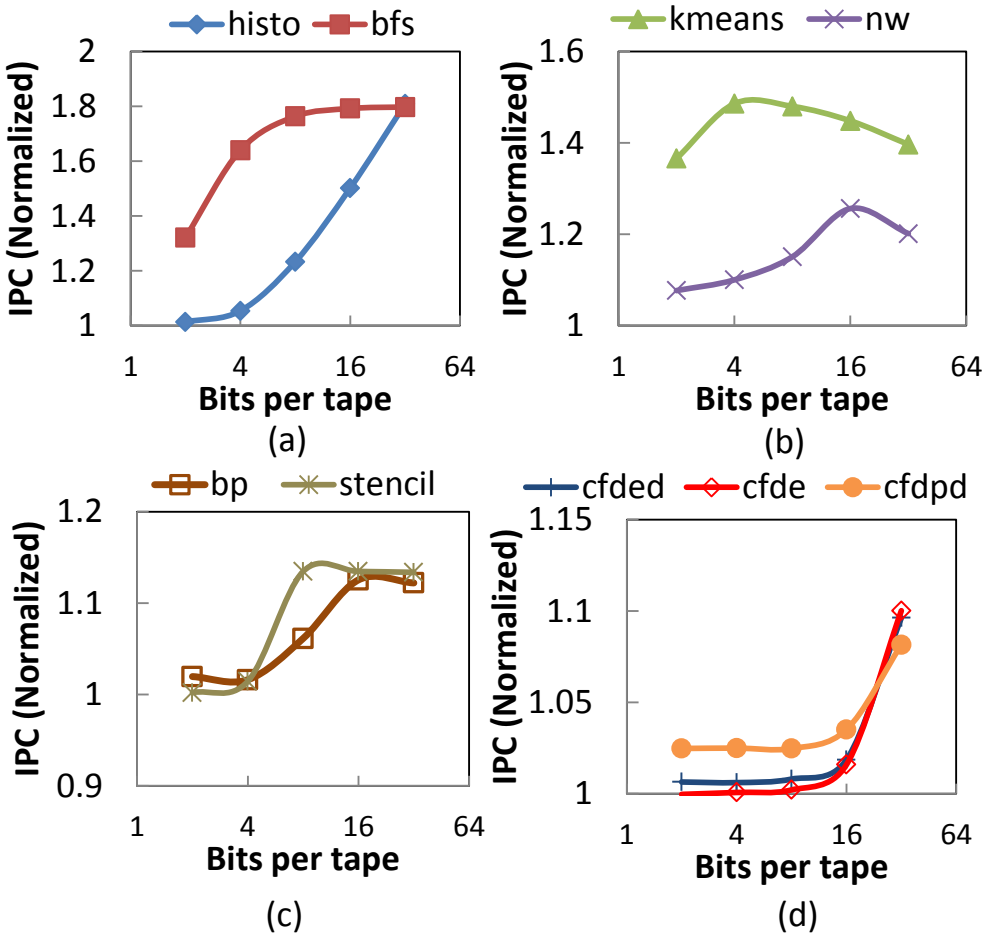


Fig. 7.9.: Impact of bits/tape on performance under iso-area conditions

First, benchmarks such as *histo* and *bfs* [Figure 7.9(a)] show monotonic increase in performance. This stems from their ability to utilize larger caches and the effectiveness of our techniques to mitigate the penalty of shifts. The second category of benchmarks [Figure 7.9(b)] show initial performance improvement, which degrades after a point. We identify two factors that influence this behavior: (i) These benchmarks (*e.g., kmeans*) are not sensitive to cache capacity once the entire working set fits in the cache and suffer small performance degradation due to increased shift latencies with larger bits per tape, and (ii) the cache management policy does not perform

well on these benchmarks (*e.g., nw*), which results in a more pronounced degradation in performance.

The third set of benchmarks like *stencil* [Figure 7.9(c)] also exhibit initial performance improvement but their IPC saturates beyond a certain bits/tape threshold. These benchmarks are relatively insensitive to cache access latencies. This observation is re-affirmed by the fact that the SaPB did not significantly improve performance. The fourth category included benchmarks like *cfde* [Figure 7.9(d)] whose IPC remained dormant at first but improved beyond a certain cache size. Analysis of these benchmarks revealed that there was significant cache thrashing, which was overcome by the increase in cache capacity. Overall, most CCS benchmarks benefited from increasing the bit count of DWM tapes. Hence, designing a dense cache by packing large number of bits is a favorable design choice for most GPGPU workloads.

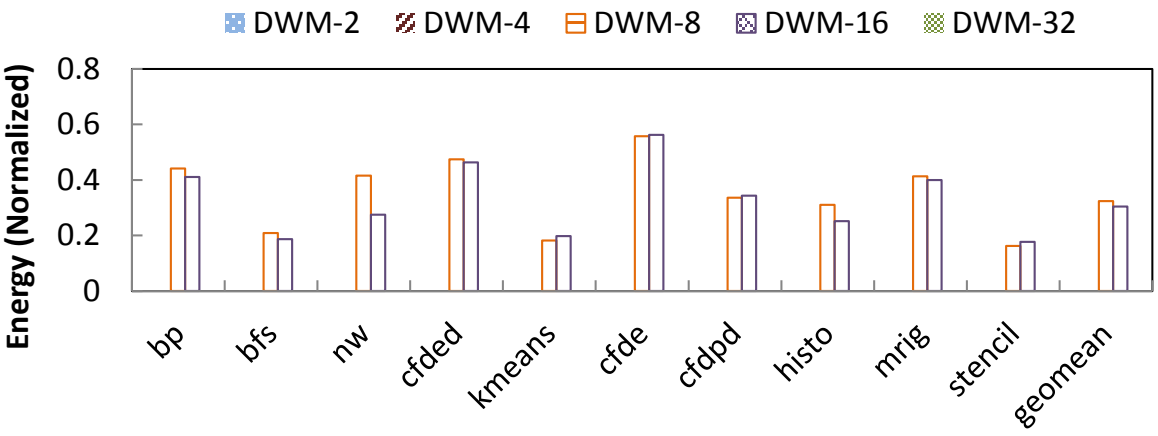


Fig. 7.10.: Energy consumption for various bits/tape configurations

We next consider the impact of increasing the number of bits per tape on energy consumption under iso-area conditions (Figure 7.10). Most CCS benchmarks exhibited a strong correlation between their energy and performance trends. This is because the reduction in off-chip accesses with increase in bits per tape plays a major role in positively impacting both their performance and energy. However, there are some outliers to this behavior. For example, in the case of *nw*, when the bits per tape is increased from 16 to 32, energy improves but performance degrades. The rea-

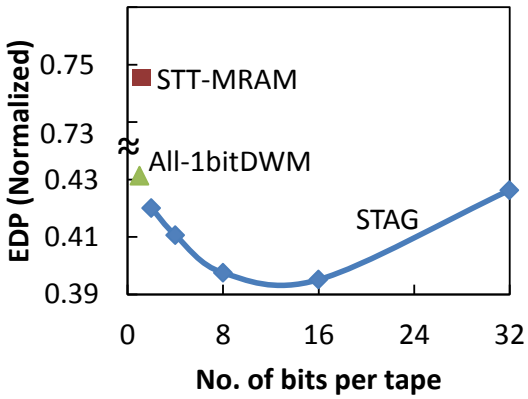


Fig. 7.11.: Impact of bits/tape on EDP

son behind this behavior is that while the decrease in off-chip accesses weighs more heavily on energy, the increased shift penalties have a higher impact on performance.

Iso-capacity design: We next consider an alternate scenario where, despite increasing the number of bits per tape, the cache capacity is maintained constant by decreasing the number of DTCs.

In this case, we observe an interesting tradeoff between energy and performance. When the bits per tape is increased, the energy consumption of the cache decreases due to lower area footprint, shorter bitlines, wordlines, I/O data lines *etc.* However, performance is adversely impacted because of the associated shift penalty. We therefore compare the Energy-Delay Product (EDP) for a cache of size 768KB for varying bits per tape in Figure 7.11. We notice that the EDP improves to reach a minimum point (around 16 bits/tape) before increasing significantly. This behavior stems from the conflicting impact on EDP due to shift penalty and reduced capacitances in the bitlines, wordlines *etc.* in the cache. Also, it is worth mentioning that across all bits/tape configurations, STAG is strictly superior to SRAM, STT-MRAM and All-1bitDWM caches.

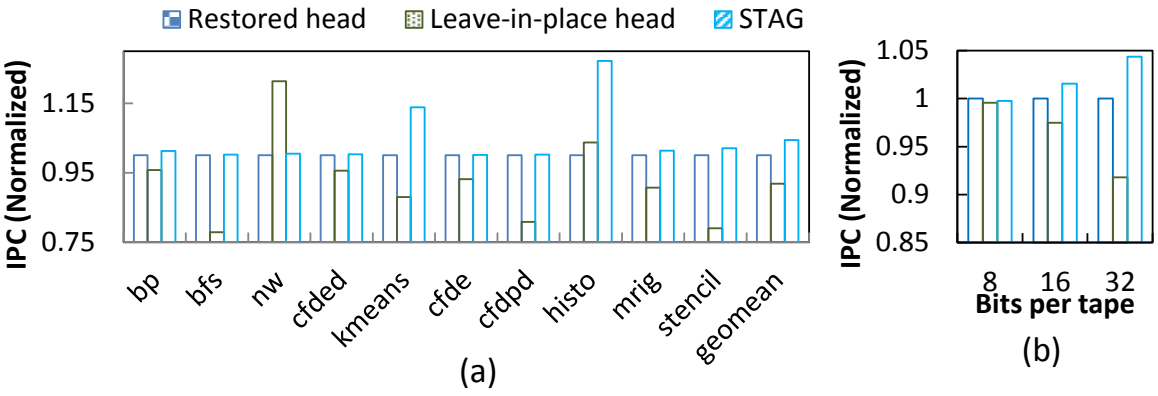


Fig. 7.12.: Impact of cache management policies on performance (Normalized to restored head policy)

Impact of cache management policies

We now study the impact of the different cache management policies on application performance. For this purpose, we implemented the restored head and leave-in-place head policies (described in Section 7.1.2) for a 768KB L2 cache containing 32 bits per tape and the application performance obtained is presented in Figure 7.12(a). We observe that STAG outperforms the restored head and leave-in-place head policies by 5.5% and 13.1%, respectively. While the restored head policy is ignorant of cache access patterns, the leave-in-place head policy suffers from interleaved accesses from several warps. Since STAG combines the strengths of both the policies, it stands out as the best design choice. Next, we evaluate the cache designs for different bits per tape in Figure 7.12(b). For fewer bits in the tape (*e.g.,* 8), the IPC remains unaffected by the choice of cache design. However, the difference becomes more pronounced as the number of bits per tape increases.

Sensitivity to cache size

We now perform a comprehensive analysis to identify the best bits/tape configuration for various cache sizes. Figure 7.13 shows a surface plot depicting the average EDP across all benchmarks for different cache sizes and bits/tape configurations. We

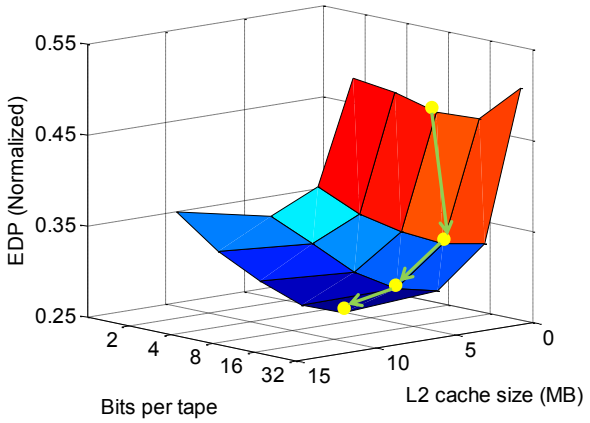


Fig. 7.13.: Impact of L2 cache size for different bits per tape

find that as the cache capacity increases, the bits/tape configuration that yields the best EDP also grows higher. For example, the optimum configuration for a 768KB L2 cache is found to be 8 bits/tape while it increases to 32 bits/tape for a 12MB cache.

7.4 Conclusion

General Purpose Graphics Processing Units (GPGPUs) have proven to be an efficient solution to accelerate a wide range of both graphics and general purpose workloads. However, a critical component for their sustained performance growth is the memory sub-system and its capability to provide data to the numerous processing cores. In this work, we propose STAG, a spintronic-tape architecture for on-chip memories in GPGPUs. STAG utilizes different DWM bit-cells to design different memory arrays in the memory hierarchy. To address the performance penalty from shifts, suitable architecture level optimizations are proposed. STAG achieves significant benefits thereby demonstrating that DWMs are highly promising candidates for designing GPGPU caches.

8. SPINTASTIC: SPIN-BASED STOCHASTIC LOGIC FOR ENERGY-EFFICIENT COMPUTING

Spin-based devices have proven to be well-suited for realizing dense, non-volatile memories with low leakage power, leading to large-scale prototypes and early commercial offerings [18,21,22]. For designing spintronic logic, various approaches have been proposed *viz.* Nano-Magnetic logic (NML) [103, 132], Domain Wall Logic (DWL) [26], mLogic [28], and All-Spin Logic (ASL) [24]. However, they are not considered to be competitive with CMOS from an energy-delay perspective [133]. For instance, ASL's speed is limited by the high current required to achieve fast non-local spin-torque switching, and its energy is limited by the short circuit power resulting from all-metallic devices and the buffers needed to overcome the limited spin-diffusion length in interconnects. On the other hand, logic styles based on magnetic switching, such as NML, incur energy for generating external magnetic fields. While materials and device structures are under active investigation to improve the competitiveness of spintronic logic, we take the complementary approach of investigating alternative computing models and application domains that match the inherent characteristics of spintronic devices.

In this work, we explore *stochastic computing* (SC) [134] as a new direction to efficiently realize logic with spintronic devices. Stochastic computing is a model of computation in which pseudo-random bitstreams are used to represent numbers, and computations are cast in terms of operations on such bitstreams. Stochastic computing, while not a general-purpose replacement for Boolean logic, has been applied to a number of prevalent and emerging application domains such as digital signal and image processing, communications, recognition, mining and synthesis [135–139].

A key characteristic of SC is that it enables compact, low-complexity logic implementations of common arithmetic operations such as addition, multiplication *etc.*

For example, a multiplier in the stochastic domain can be realized with only an AND gate, as the probability of '1' at the output of an AND gate is the product of the corresponding probabilities at its inputs. However, this reduction in complexity comes at a cost. SC requires additional circuits to convert data between stochastic and binary number representations and to eliminate correlations across bitstreams. These peripheral circuits present significant overheads — often consuming 80-90% of the power in CMOS implementations — and severely limit the overall energy efficiency of SC. Another challenge to the efficiency of SC is that the length of the stochastic bitstream grows exponentially with the desired degree of precision. For example, uniquely representing 8-bit data in SC requires bitstream of length 256 (2^8). Since the number of compute cycles is proportional to the bitstream length, longer bitstreams lead to significant degradation in the performance of SC implementations.

The primary contribution of this work is to demonstrate *the synergy between spintronic devices and stochastic logic*. On the one hand, the physical characteristics of spin devices can be exploited to efficiently realize the key components of stochastic logic circuits. On the other hand, the low complexity and bit-level parallelism of stochastic logic circuits can alleviate the shortcomings of spin-based logic. Based on the above insight, we present SPINTASTIC, a new approach for the energy-efficient implementation of stochastic logic circuits using spintronic devices.

We make the key observation that the physical characteristics of spintronic devices can be exploited to efficiently design the peripheral circuits required for SC. It has been demonstrated that, with careful design, a nanomagnet can act a random number generator (RNG) [140], and produces pseudo-random bits due to the induced thermal instability. Using this behavior, we design a spin-based stochastic bitstream generator that is highly efficient compared to its CMOS counterpart. In addition, we also propose a spin-based stochastic bitstream permuter that eliminates correlation across stochastic bitstreams. Along with the Spin-RNG, the permuter utilizes the property of domain wall motion in ferromagnetic nanowires to efficiently shuffle bits within a stochastic bitstream. To design the various arithmetic blocks of SC circuits,

we utilize All-Spin Logic (ASL) [24]. The bit-level parallelism in SC along with the non-volatility of ASL gates is exploited to realize fine-grained pipelined implementations, which leads to improvements in throughput and energy efficiency. Further, we exploit the high density of spintronic building blocks to realize vectorized-SPINTASTIC design that processes multiple bits of a given stochastic bitstream in parallel, thereby significantly reducing the number of compute cycles required for stochastic processing. It is noteworthy that, while this optimization can be performed in the context of CMOS SC implementations, it is more pronounced with spintronic implementations as spintronic designs are extremely compact.

In summary, the contributions of the work are as follows:

- We identify the synergy between stochastic computing and spintronic devices, and propose SPINTASTIC, an energy-efficient spintronic logic design paradigm.
- We exploit the physical characteristics of spintronic devices to efficiently design the key building blocks required for stochastic processing *viz.* stochastic arithmetic units, stochastic number generator, stochastic bitstream permuter and stochastic-to-binary converter.
- We explore different optimizations – fine-grained pipelining and vector processing – that exploit the bit-level parallelism offered by SC to enhance the benefits of SPINTASTIC.
- To evaluate the benefits of SPINTASTIC, we develop a rigorous modeling/simulation framework consisting of physics-based Landau-Lifshitz-Gilbert (LLG) [141] and modified Valet-Fert [142] models to accurately capture the magnetization dynamics and transport characteristics of spintronic devices. Our experiments on 8 benchmark circuits and 3 popular applications from recognition, mining, synthesis (RMS), signal processing, and image processing application domains demonstrate significant improvements in energy over a well-optimized 16nm CMOS baseline.

The rest of the chapter is organized as follows. Section 8.1 provides relevant background on stochastic computing. Section 8.2 explains the spin device concepts used in SPINTASTIC. Section 8.3 describes the design of various components SPINTASTIC and Section 8.4 presents the vectorized-SPINTASTIC design. Section 8.5 presents the physics-based modeling framework employed for evaluating SPINTASTIC and the results are presented in Section 8.6. Section 8.7 concludes the chapter.

8.1 Background: Stochastic Computing

Stochastic computing is a probabilistic model of computation originally proposed in [134], in which data is represented and processed in the form of pseudo-random bitstreams. While early research in SC focused on the number representation and realization of basic arithmetic operations, subsequent research efforts have demonstrated SC implementations for a wide range of applications domains such as image processing [135, 136], neural networks [137], signal processing [138], and recognition, mining, and synthesis (RMS) [139].

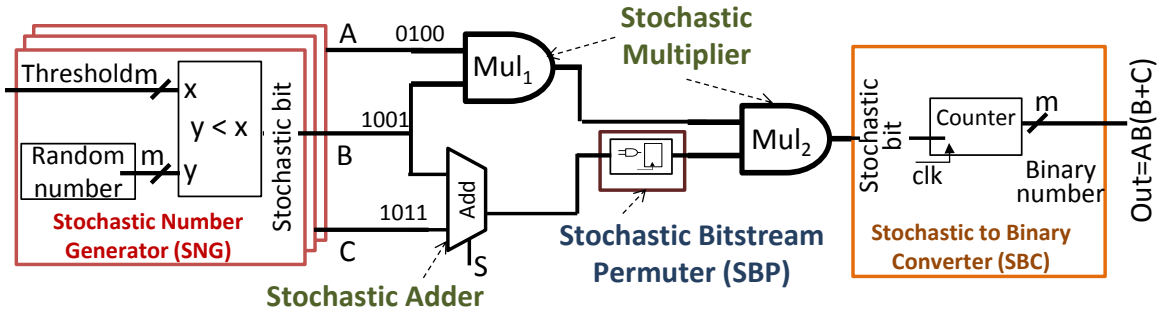


Fig. 8.1.: Structure of a stochastic computing circuit

Figure 8.1 describes the structure of a typical stochastic computing circuit. First, a stochastic number generator (SNG) block is used to generate stochastic bitstreams of the desired probability. This is achieved by comparing a series of random numbers generated by a pseudo-random number generator circuit (*e.g.*, Linear Feedback Shift Register) with the binary number to be represented. Next, the stochastic bitstreams

are processed by different stochastic arithmetic units that are connected together to realize the desired function. The example in Figure 8.1 shows stochastic implementations of two most commonly used arithmetic units—multiplier and adder. A stochastic multiplier can be realized by just an AND gate, as the probability of '1' in the bitstream at the output of the AND gate is the product of the probabilities of the bitstreams at its inputs. An adder in stochastic domain is realized using a MUX whose inputs are the bitstreams to be added and whose select signal is fed a random bitstream of probability 0.5. The probability of '1' at the output of the MUX is the average of its inputs, thus achieving scaled addition. We note that stochastic circuit implementations of other complex blocks such as exponential, log *etc.* have been proposed, and stochastic computing has been extended with the ability to represent and process negative numbers [134]. A key requirement for a stochastic computation to yield correct outputs is that the different pseudo-random bitstreams processed together are uncorrelated. At the inputs, this can be achieved by seeding the random number generators differently. However, to avoid correlations when bitstreams re-converge, stochastic bitstream permuter (SBP) circuits are employed (as shown in Figure 8.1 at one of the inputs of Mul_2) to randomly shuffle stochastic bitstreams, thereby eliminating their correlation. In CMOS implementations, an SBP is realized using the SBC and SNG circuits back-to-back *i.e.* by first converting the stochastic bitstream to binary and then regenerating the bitstream. Finally, a stochastic-to-binary converter (SBC) circuit produces the output in binary format. This is achieved by using an up-counter that counts the '1's in the output bitstream to ascertain its magnitude.

From the above discussion, it is evident that although stochastic arithmetic units are compact, the peripheral circuits *viz.* SNG, SBC, and SBP, dominate the energy consumption of SC, severely limiting its overall energy efficiency. For example, the peripheral circuits accounted for over 89% of the total energy in our stochastic implementation of a 1D-DCT circuit.

8.2 SPINTASTIC: Device fundamentals

SPINTASTIC uses three key concepts from spintronic device literature—spin random number generator (Spin-RNG), All-Spin Logic (ASL), and domain wall motion. The concepts of ASL and domain wall motion were introduced earlier in Chapter 3. In this section, we present a brief description of spintronic random number generator.

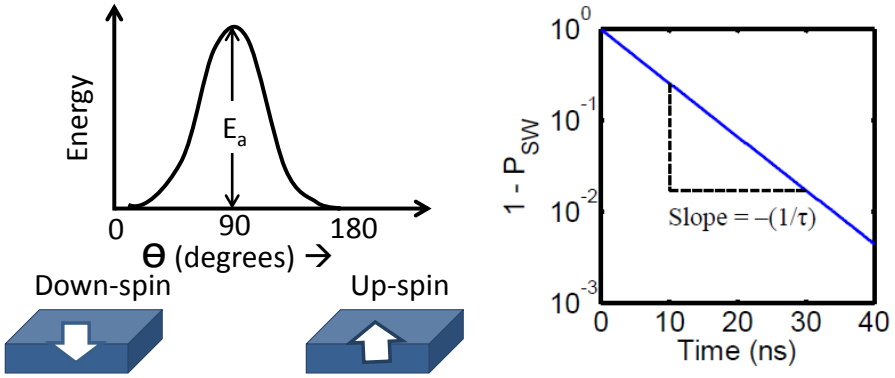
Spin random number generator: A spin random number generator (Spin-RNG) [140] is designed by exploiting the naturally occurring random switching of a nanomagnet. A nanomagnet, due to its shape anisotropy, exhibits two stable states that are separated by an energy barrier (E_a) as shown in Figure 8.2a. If the energy barrier is sufficiently lowered, it leads to thermal instability within the nanomagnet and it periodically flips its magnetization orientation due to thermal noise. The period between successive flips, referred to as the characteristic switching time (τ), is given in Equation 8.1.

$$\tau = t_0 e^{(E_a/k_B T)} \tag{8.1}$$

In Equation 8.1, t_0 is the attempt period (typically 1ns), k_B is the Boltzmann constant and T is the absolute temperature. Recent efforts on Spin-RNG have demonstrated switching times as low as 10s of nanoseconds and developed mechanisms to read the random state of the nanomagnets without influencing the same [140]. In SPINTASTIC, we employ Spin-RNG to generate, as well as, permute stochastic bitstreams of any desired probability. Based on the switching probability profile (Figure 8.2b), we operate the Spin-RNG with a time period of 20ns. We note that, *unlike cryptographic applications, high-quality random numbers are not required for SC* — this is leveraged even in CMOS implementations, were LFSRs are typically used [135–139].

8.3 SPINTASTIC logic design

In this section, we present the design of four key components, *viz.* spintronic stochastic number generator, spintronic stochastic bitstream permuter, spintronic



(a) Energy landscape of a nanomagnet (b) Switching probability vs. time [140]

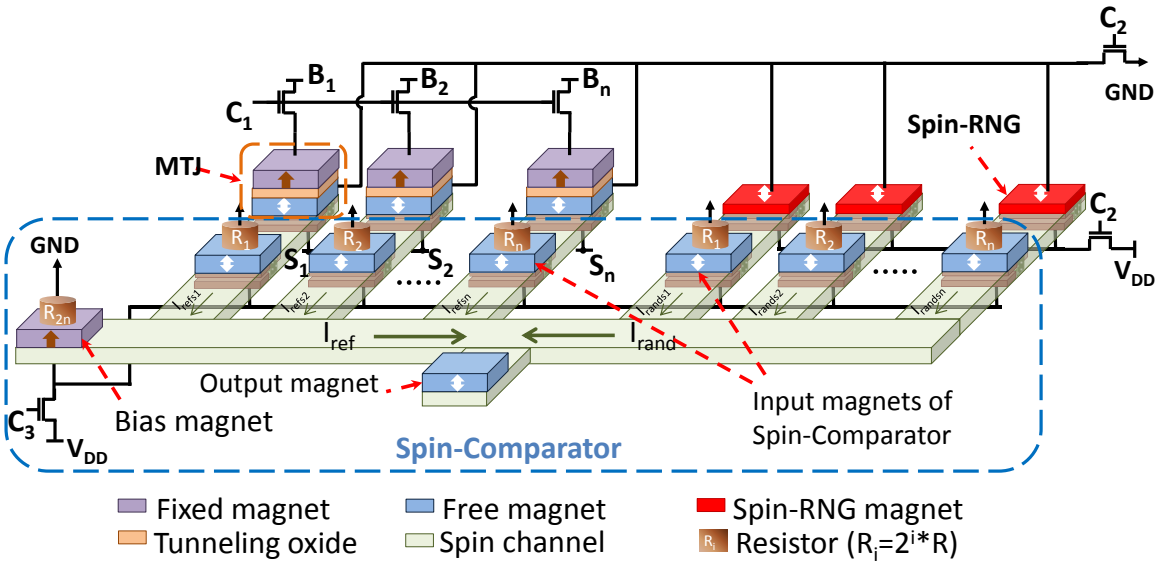
Fig. 8.2.: Spintronic Random Number Generator (Spin-RNG)

stochastic-to-binary converter, and spintronic stochastic arithmetic units, which form the fundamental building blocks of SPINTASTIC.

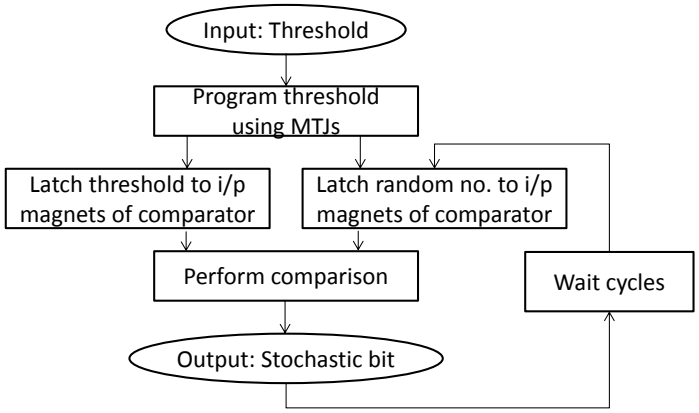
8.3.1 Spintronic Stochastic Number Generator

As described in Section 8.1, the Stochastic Number Generator (SNG) is responsible for converting a binary number into an equivalent stochastic bitstream. In a typical SNG, each bit of the stochastic bitstream is generated by comparing the binary number with a pseudo-random number. If the binary number is greater, then the stochastic bit is set to '1', else it is set to '0'. Thus, the number (probability) of ones in the stochastic bitstream is directly proportional to the binary number.

In SPINTASTIC, we utilize the concept of Spin-RNGs, described in Section 8.2, to design the spintronic stochastic number generator (Spin-SNG). Figure 8.3 shows the structure of the Spin-SNG. It consists of a set of Spin-RNG magnets that independently generate bits of the random number. An array of magnetic tunneling junctions (MTJs), a structure formed by two ferromagnets (a fixed magnet and a free magnet) that are separated by a tunneling oxide, is used to store the binary number. Any binary number can be programmed into the MTJs by driving nodes B_i and S_i in Figure 8.3 to appropriate voltages. Next, the random number and threshold are



(a) Schematic



(b) Operation

Fig. 8.3.: Spintronic Stochastic Number Generator (Spin-SNG)

latched into the input magnets of a *spin comparator*. The spin comparator operates by injecting a current through each input nanomagnet that is proportional to the place value of the bit that it stores. As illustrated in Figure 8.3, this is achieved by placing resistors of exponentially varying magnitudes (R_i=$2^i * R$) above the spin comparator's input nanomagnets. As a result, spin-polarized currents are injected into the channel, and the net polarization of the spin current in the channel is determined by the relative magnitudes of the random and binary numbers. The spin polarization in the channel switches an output magnet using the phenomenon of non-local spin

torque, to produce the stochastic bit. Note that, in Figure 8.3, a bias magnet with fixed magnetization is also connected to the spin channel to ensure that the output magnet of the comparator flips to a stable value when the binary and the random numbers are of equal magnitude.

The different steps in the operation of the Spin-SNG are shown in Figure 8.3b. First, the binary number is programmed using the MTJs by driving C_1 to V_{DD}, and nodes B_i and S_i (i=1 to n) to appropriate voltages. For example, to program '0' ('1'), B_i is connected to V_{DD} (GND) and S_i to GND (V_{DD}). Next, the bits generated by the Spin-RNG magnets are latched to the input of spin comparator by connecting C_2 to V_{DD} and injecting currents of appropriate magnitude through them. After this, the spin comparator is activated by driving C_3 to V_{DD} and the result of the comparison is produced at the output. This process is repeated for a sequence of random numbers generated by the Spin-RNGs until the stochastic bitstream of the desired length is obtained. Also, in order to ensure that the outputs of the Spin-RNG magnets are random, a wait time greater than the characteristic switching time of the Spin-RNG is introduced between successive stochastic bit outputs.

Design constraints and challenges: The Spin-SNG imposes several design requirements for proper generation of the stochastic bitstream. A key constraint in the design of the Spin-SNG is the wait time between two consecutive stochastic bit outputs. As described in Section 8.2, a Spin-RNG would require a wait time of about 20ns. As a result, the operating frequency limited and can lead to significant overheads in both performance and energy. In order to address this issue, we utilize a *multiplexed stochastic number generator*. We exploit the compactness of Spin-SNGs and utilize multiple instances thereof to generate different stochastic bitstreams of the same probability. These stochastic bitstreams are then time multiplexed in a round-robin manner to generate a single stochastic bitstream at a higher throughput.

Another major challenge in the design of Spin-SNG is the restriction on fan-in of the spin comparator that is imposed by the limited spin diffusion length. For proper functioning of Spin-SNG, the distance between the input and output magnets needs

to be smaller than the spin diffusion length. As the number of fan-in increases, the distance between the input and output magnets increases, which places constraints on the maximum fan-in of Spin-SNG. Fortunately, most of the applications typically have smaller precision (bit-width is usually ≤ 8 bits) and therefore, the maximum fan-in of the spin comparator is low enough to be realized within the constraints imposed by spin diffusion length. Further, if required, spin comparators of higher precision can be easily realized by composing multiple low precision comparators by first comparing the MSB and LSB parts separately and then combining the results in a hierarchical fashion.

8.3.2 Spintronic Stochastic Bitstream Permuter

The objective of the stochastic bitstream permuter circuit is to shuffle the bits of a stochastic bitstream $(b_0 \ldots b_{N-1})$ in order to eliminate correlation. Figure 8.4 shows the design of the Spintronic Stochastic Bitstream Permuter (Spin-SBP). It consists

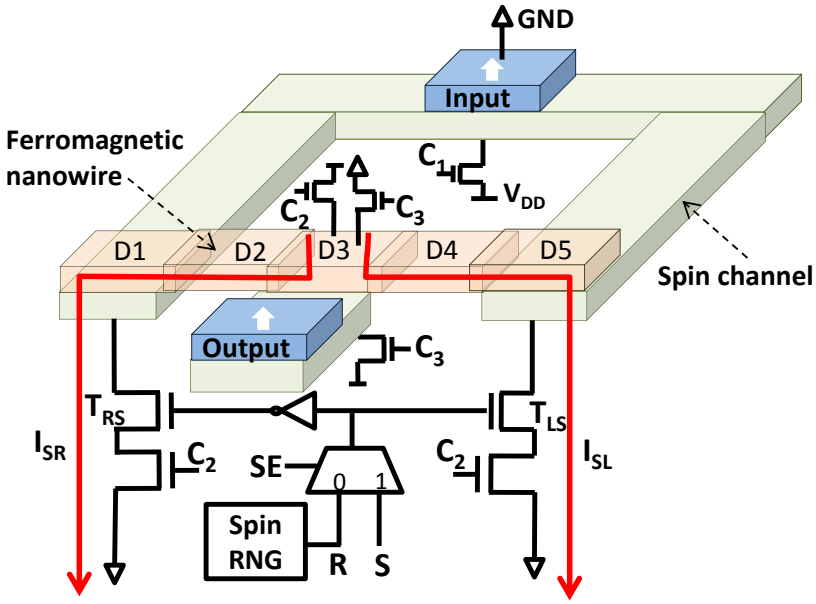


Fig. 8.4.: Spintronic Stochastic Bitstream Permuter

of an input nanomagnet, an output nanomagnet, a ferromagnetic nanowire with 5

magnet domains, a Spin-RNG with peripheral circuitry, and NMOS transistors. The input nanomagnet is connected to the two end domains of the nanowire (domains $D1$ and $D5$) and the output nanomagnet is connected to domain D3 using spin channels. The domains $D1$ and $D5$ are also connected to the drain terminal of the T_{RS} and T_{LS} transistors respectively. As shown in Figure 8.4, the gate terminal of the transistors are connected to the random bit output R, after feeding it through a low-overhead control logic comprising of 2 control inputs *viz.* Set(S), Set Enable(SE). This arrangement facilitates the domains in the nanowire to be partially shifted in either direction based on the random bit output. For example, when the random bit output is 0 ($R = 0$) and the control inputs are set to appropriate values ($SE = 0$, $S = X$,$C_2 = 1$), the transistor T_{RS} is turned ON and a current (I_{SR}) flows from the middle domain ($D3$) of the nanowire through the domains to the left of it *viz.* $D1$ and $D2$. Due to the phenomenon of domain wall motion, the bits stored in the domains $D1$ and $D2$ shift to the right *i.e.* the magnetic orientation of D2 is shifted to $D3$, while that of $D1$ is propagated to $D2$. Similarly, when the random bit is 1 ($R = 1$, $SE = 0$, $S = X$, $C_2 = 1$), the domains $D4$ and $D5$ are left shifted due to the flow of current I_{SL} in the right half of the nanowire. Thus, based on the random bit output, the orientation of either $D2$ or $D4$ can be propagated to $D3$, which is subsequently read at the output of the Spin-SBP by asserting C_3.

The operation of the Spin-SBP is explained below. The Spin-SBP takes one bit of the stochastic bitstream as its input and produces one bit of the permuted bitstream at the output in each cycle of operation. First, each input bit to the Spin-SBP is latched to both the ends of the DWM nanowire (domains D1 and D5) through the spin channels by connecting C_1 to V_{DD}. The following control sequence is carried out in the first two execution steps. In the first execution step, after latching the first bit to end domains (D1 and D5), the control inputs C_2, S and SE are asserted and therefore T_{LS} is turned ON irrespective of the random bit output. This results in a left shift of the domains in the nanowire and the first stochastic bit input (b_0) is propagated to domain $D4$. In the second execution step, the second stochastic bit

input (b_1) is first stored in the end domains and then shifted to domain $D2$ by setting the control inputs to $C_2 = 1$, $S = 0$ and $SE = 1$. From the third step onwards, we latch the input bit to domains D1 and D5, and then set $C_2 = 1$, $S = 0$, and $SE = 0$. Therefore, the random bit output (R) determines the direction in which the bits in the nanowire is shifted. For example, if $R = 0$ in the third cycle, the stochastic input bit b_1 is shifted to domain $D3$, which is subsequently produced as the first shuffled bit output. Consequently, because of the right shift, the third stochastic bit latched in the domain $D1$ replenishes $D2$. Therefore, the next shuffled bit output could be either b_0 or b_2 depending on the random bit output. This process is repeated until $N - 2$ shuffled output bits are produced by the Spin-SBP. Then the control sequence described above for the first two cycles of operation is repeated to read out the last two output bits.

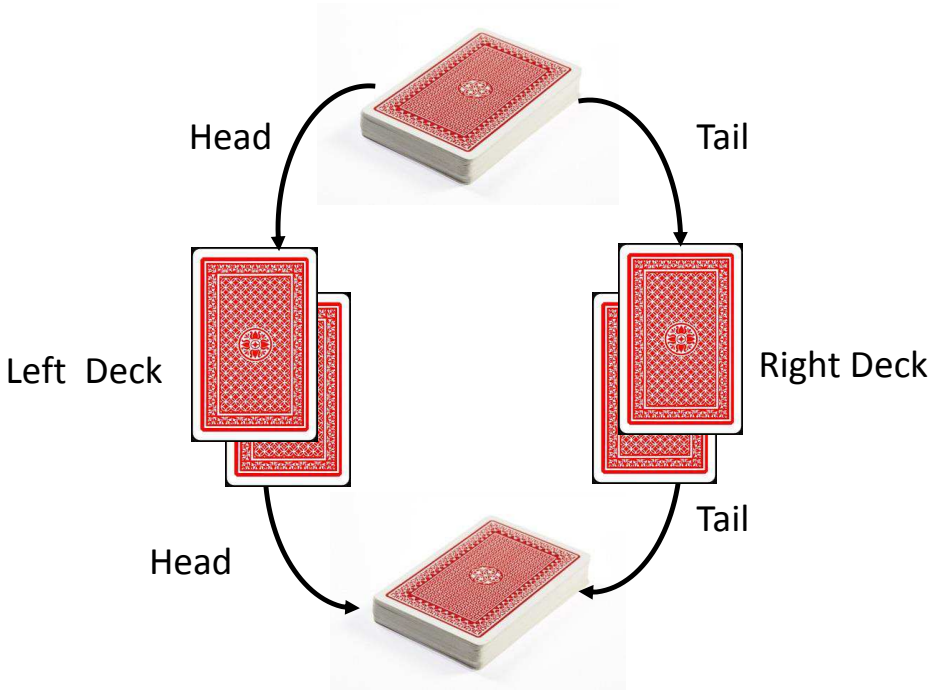


Fig. 8.5.: Logical view of Spintronic Stochastic Bitstream Permuter

This randomization process can be thought analogous to shuffling a deck of cards using a toss of coin as shown in Figure 8.5. If the coin toss results in a 'Head', then

the card at the top of the deck is placed on the top of the left deck and the card at the bottom of the left deck is moved to the bottom of output deck. Similarly, if the coin toss results in a 'Tail', the card at the top of the input deck is placed on the top of the right deck and that at its bottom is moved to the bottom of output deck. This process is repeated until all the cards in the input deck is moved to the output deck.

The proposed Spin-SBP has a number of advantages. Since, each bit (b_i) of the stochastic bitstream can occupy a position between $i-1$ and $N-1$ in the shuffled bitstream, the Spin-SBP can produce $N(N!)$ permutations of the possible N^N permutations of the input bitstream. The number of the possible permutation outcomes can be further improved by having domain wall tapes of larger length. In contrast, a LFSR-based implementation can produce only N different shuffled bitstreams based on the value of its seed. Another key benefit of the proposed SBP is that it is extremely compact and is based on highly energy efficient domain wall motion [21]. In comparison, a conventional CMOS design of a permuter involves a combination of an SBC and an SNG, leading to large area and energy overheads. Further, since the proposed SBP does not involve stochastic-to-binary conversion, it does not require all the bits of the stochastic bitstream to be available before generating the first output bit, which leads to improvement in performance.

8.3.3 Spintronic Stochastic-to-Binary Converter

The Spintronic Stochastic-to-Binary converter (Spin-SBC) is responsible for translating the stochastic bitstreams into their binary equivalents. This involves counting the number of 1s in the stochastic bitstream, thereby inferring its magnitude. In SPINTASTIC, we design the Spin-SBC using the ASL-based functionality enhanced spin-full adder cells proposed in [104]. Figure 8.6 shows the design of the Spin-SBC, which consists of multiple spin-full adder cells chained together in a ripple-carry fashion. This design of Spin-SBC would be energy inefficient compared to its CMOS counterpart as it possesses high logic depth. For example, even accumulating a bit-

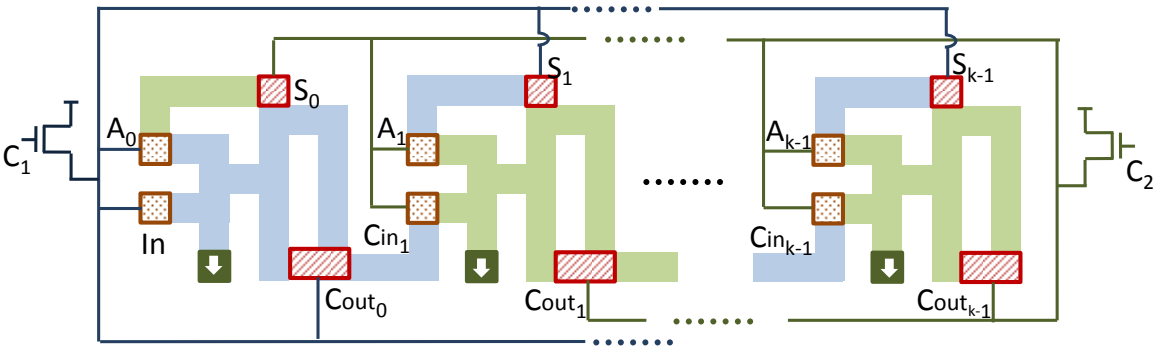


Fig. 8.6.: Spintronic Stochastic-to-Binary Converter (Spin-SBC)

stream of length 256 requires a Spin-SBC containing 8 levels of logic. To improve the efficiency of the Spin-SBC, we exploit the parallelism between adjacent bits in the stochastic bitstream and the non-volatility of nanomagnets to operate the different stages (spin-full adder cells) of the Spin-SBC in a pipelined fashion. To ensure correct functionality under pipelined operation, the inputs to a given stage of the pipeline needs to be held constant when it is being evaluated. However, nanomagnets are inherently level-sensitive devices *i.e.* their orientation can be flipped at any point during the course of their evaluation. Therefore, adjacent stages of the pipeline cannot be operated together in the Spin-SBC. To address this, we employ two non-overlapping control signals $C1$ and $C2$ that are connected to the odd and even stages of logic as shown in Figure 8.6. The stages are evaluated only when the control signal connected to them are asserted. Thus odd and even stages of the pipeline operate exclusively and a K-Stage Spin-SBC can process K/2 stochastic bits in a pipelined manner.

8.3.4 Spintronic Stochastic Arithmetic Units

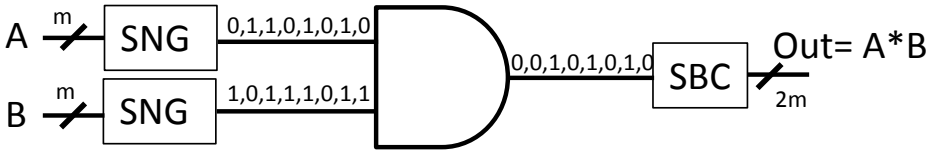
The Spintronic Stochastic Arithmetic Units (Spin-SAU) perform the designer computation on the stochastic bitstreams generated by Spin-SNG. In SPINTASTIC, we design different spintronic arithmetic units using ASL described in Chapter 3. ASL logic implementations for simple Boolean logic gates (*e.g.,* AND, MUX, *etc.*) that are used to perform arithmetic computations on stochastic bitstreams are available

in literature [104] and can be directly employed in SPINTASTIC. Further, we exploit the bit-level parallelism across bits in the stochastic bitstream and the non-volatility of nanomagnets to design fine-grained pipelined implementations of stochastic arithmetic units to improve the throughput.

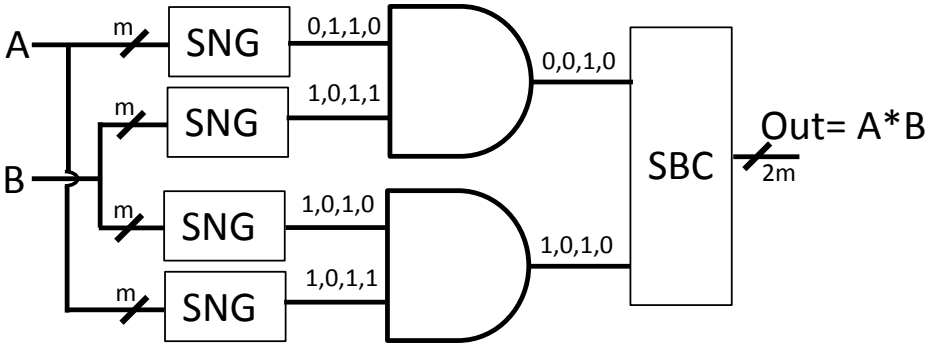
Thus, in SPINTASTIC, we utilize the physical characteristics of the spin devices to efficiently design the different components of stochastic logic circuits.

8.4 Vectorized-SPINTASTIC logic

One of the major benefits of SPINTASTIC logic is the reduction in area achieved through both the high density of spintronic building blocks and the low logic complexity of SC. Our analysis across a wide range of benchmarks shows that SPINTASTIC achieves 5X-17X (10X average) reduction in area compared to binary CMOS implementation. In order to translate this area reduction into energy and performance benefits, we propose vectorized-SPINTASTIC logic design in which we re-invest the area savings to increase the number of SPINTASTIC logic units and process the bits of the stochastic bitstreams in parallel.



(a) Without vectorization



(b) With vectorization

Fig. 8.7.: Stochastic multiplier design

Figure 8.7 demonstrates the concept of vectorization [139] for a stochastic multiplier design. In a typical stochastic multiplier without vectorization (Figure 8.7a), each bit of a stochastic bitstream of length 'n' (say) is processed in a sequential fashion. This results in significantly higher latencies and places stringent design constraints on the spintronic logic gates in terms of high switching currents for fast operation. This overhead can be addressed through vectorization shown in Figure 8.7b that employs multiple stochastic logic units, each of which operate on a stochastic bitstream of shorter length in parallel. In general, for a vectorization factor of 'k', we employ 'k' stochastic units (SNGs, SAUs *etc.*) and each stochastic unit operates on bitstream of length n/k, leading to 'k' times reduction in the latency of operation. This significantly relaxes the delay requirements of each spintronic logic unit, thereby enabling its operation at relatively lower current values. This in turn leads to significant improvement in the energy consumption of SPINTASTIC design. In addition, vectorization also augments the scope for other spintronic logic optimizations like stacking [113] in which multiple spintronic logic units can be stacked to share the common current source. In other words, the overhead of introducing multiple spin-based vector processing units can be greatly amortized, leading to further improvements in energy consumption.

Note that, the concept of vectorization can also used to enhance the benefits of stochastic CMOS designs. However, vectorization requires multiple peripheral units like SNGs, SBPs *etc.*, which dominate the energy consumption and area of stochastic CMOS implementations. We observe that stochastic CMOS achieves only $\sim 4X$ area reduction across different benchmark circuits, which greatly limits the vectorization factor and therefore, the benefits of vectorization.

8.5 Experimental methodology

SPINTASTIC modeling framework: In order to evaluate the proposed SPINTASTIC design, we have developed a systematic physics-based device simulation frame-

Table 8.1.: Benchmark circuits and applications

Circuits	32-Tap FIR filter, 8-input 1D DCT, 8-input 2D DCT, 16-point FFT, Euclidean distance (EUD), sum of absolute difference (SAD), perceptron classifier (PC), artificial neural network (ANN)
Applications	Support Vector Machines (SVM), k-Nearest Neighbors (kNN), Generalized Learning Vector Quantization (GLVQ)

work for different spintronic building blocks. Our framework consists of two self-consistent procedures: (i) spin transport simulation based on modified Valet-Fert diffusion model [142], and (ii) Landau-Lifshiftz-Gilbert (LLG) equation for capturing the magnetization dynamics of spin-torque switching in a nanomagnet [141]. This framework was used to evaluate the switching current and delay for ASL [143] as well as the shift current and shift time for domain wall motion [144]. In order to model the switching current and timing requirements of an MTJ, we used NEGF to calculate current-voltage characteristics and spin-current dependent torque of MTJ, which is then used by LLG to simulate the magnetization dynamics [145]. The Spin-RNG was evaluated using the model proposed in [140]. The device parameters for our experiments were chosen based on [140, 144, 145]. The device level characteristics were then incorporated into an analytical model to compute the total delay and energy consumption of the different SPINTASTIC designs.

CMOS baselines and benchmark circuits: In this work, we compare SPINTASTIC and vectorized-SPINTASTIC designs with three different CMOS baselines—Binary CMOS (B_CMOS), Stochastic CMOS (St_CMOS), and Stochastic CMOS with vectorization (St_CMOS+Vec). The benchmark circuits were implemented at the Register-Transfer Level (RTL) and synthesized to the 16nm technology library using Synopsys Design Compiler. Synopsys Power Compiler was then used to estimate the power, delay, and energy consumption of the benchmarks. In our evaluation, we considered 8 benchmark circuits, shown in Table 8.1, from the signal processing, image processing and Recognition, Mining and Synthesis (RMS) application domains. In addition to the aforementioned data-path modules, we also evaluate the energy

benefits of SPINTASTIC at the application level for three different applications listed in Table 8.1. For all the benchmarks, we assume a bit-width of 8 bits for the CMOS binary implementation and an equivalent bitstream of length 256 bits for the stochastic designs.

8.6 Experimental results

In this section, we present the results comparing the energy consumption of SPINTASTIC with CMOS binary and stochastic implementations.

8.6.1 Energy benefits and analysis

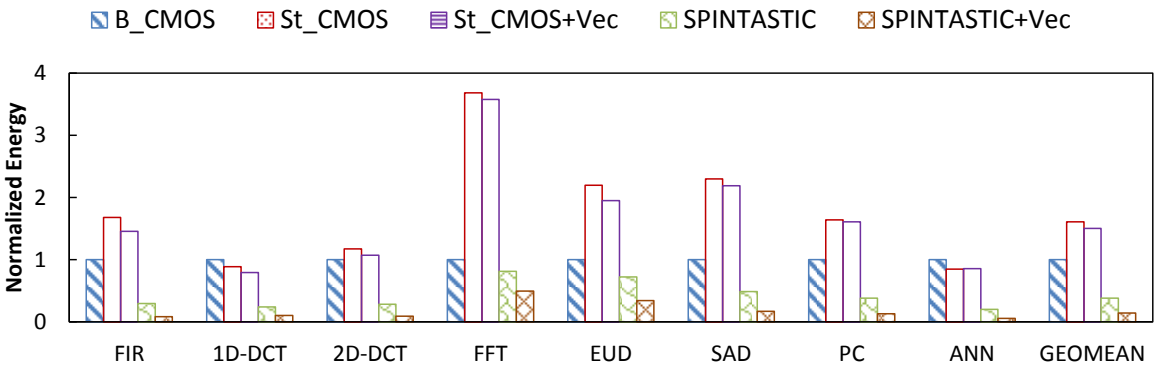


Fig. 8.8.: Comparison of energy consumption of SPINTASTIC with CMOS baseline designs

Figure 8.8 presents the normalized energy benefits of SPINTASTIC over the CMOS baselines. Compared to binary CMOS (B_CMOS), SPINTASTIC achieves between 1.2X - 4.95X (2.6X average) reduction in energy. The benefits stem from three key factors: (i) The intrinsic advantages of spintronic devices such as non-volatility and low operating voltage (25 mV), (ii) The low logic complexity (and logic depth) of stochastic computing circuits, and (iii) The ability to efficiently implement the peripheral circuits associated with SC using SPINTASTIC. The energy benefits of SPINTASTIC are more pronounced, to the tune of 3X - 5.7X (4.2X average), when compared

to the stochastic CMOS (St_CMOS) baseline. This is because, the CMOS stochastic baselines were energy inefficient compared to the CMOS binary implementations due to the large energy overheads associated with the peripheral circuits (SNG, SBP, and SBC). This underscores the synergy between stochastic computing and spintronic devices. The benefits of SPINTASTIC get significantly enhanced when we optimize the design using vectorization (SPINTASTIC+Vec in Figure 8.8), leading to 7.1X reduction in energy over B_CMOS and 10.7X reduction over stochastic CMOS with vectorization (St_CMOS+Vec). Note that, the benefits of vectorization are much more significant with SPINTASTIC as compared to the stochastic CMOS implementations. This is primarily due to two reasons: (i) Higher density of SPINTASTIC enables higher vectorization factor compared to stochastic CMOS, and (ii) Increased peripheral circuitry overhead with vectorization amortizes the benefits in the case of stochastic CMOS designs.

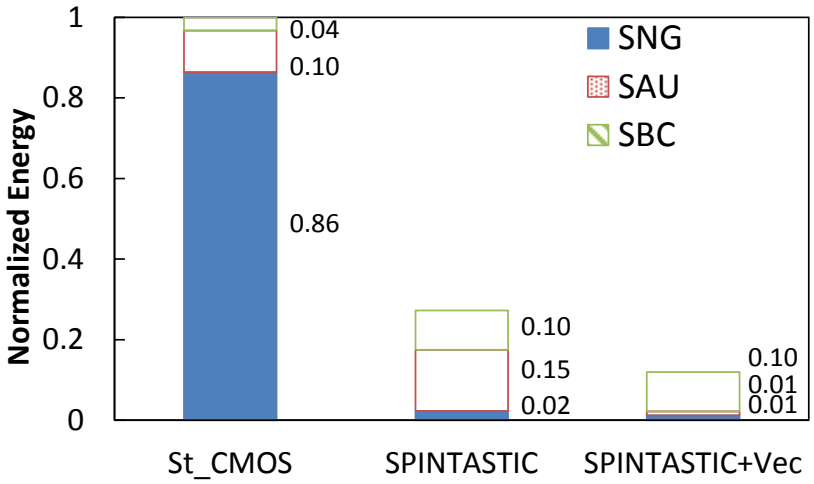


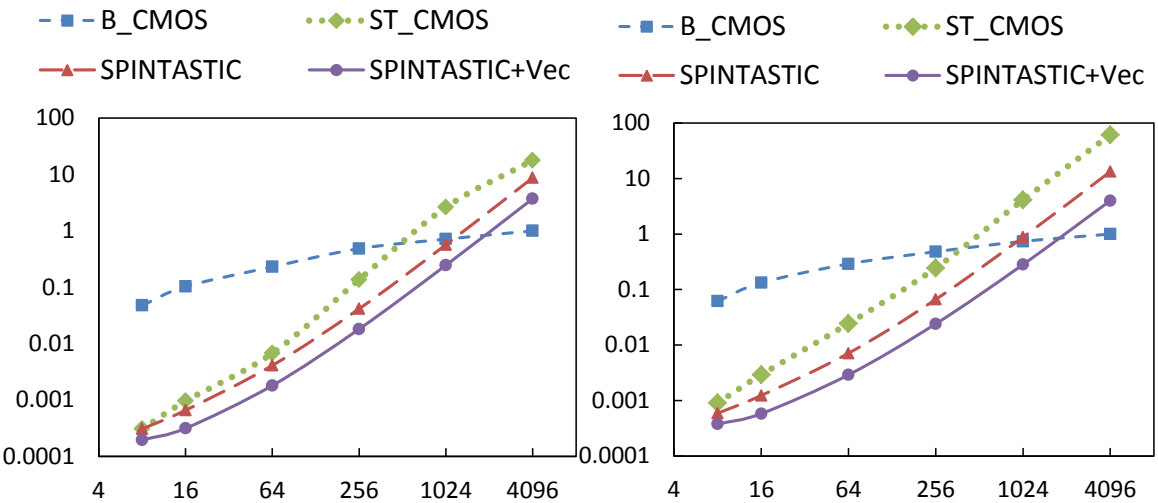
Fig. 8.9.: Energy breakdown for 1D-DCT

To better understand the sources of energy benefits in SPINTASTIC, Figure 8.9 provides the energy consumed by the different components in stochastic CMOS and SPINTASTIC designs for the 8-input 1D-DCT benchmark. The 1D-DCT benchmark uses 72 SNGs, 8 SBCs, 64 stochastic multipliers and 8 stochastic adders. As shown in Figure 8.9, the SNGs dominate the energy consumption, contributing ~86% of

the overall energy, while the stochastic multipliers and adders only consume ∼11% of the energy. This is attributed both to the number of SNGs in 1D-DCT and the high complexity of Linear Feedback Shift Registers (LFSRs) used in their design. SPINTASTIC addresses the dominant component of the energy consumption, as the proposed Spin-SNGs are 43X more energy efficient compared to CMOS SNGs. This translates to $\sim 4X$ improvement in the overall energy. Figure 8.9 also presents the energy breakdown for SPINTASTIC design with vectorization. Note that, vectorization leads to a very high reduction in the energy consumption of SAU (the dominating component of energy in SPINTASTIC), resulting in more than 8X improvement in the total energy consumption compared to the stochastic CMOS implementation.

8.6.2 Sensitivity to bitstream length

In order to study the impact of computation precision on the energy consumption of SPINTASTIC, Figure 8.10 compares the energy benefits at different bitstream lengths (8-4096 bits) for two benchmarks–1D-DCT and Perceptron. In this case, the



(a) 8-input 1D-DCT (b) Perceptron

Fig. 8.10.: Impact of bitstream length on the energy consumption

binary CMOS baseline circuits were re-synthesized at different bit-widths (3-12 bits)

corresponding to the length of the stochastic bitstream. In the case of stochastic implementations (SPINTASTIC as well as St_CMOS), we find that the energy decreases linearly (both X and Y axes are in log scale) as the bitstream length is decreased. For low bitstream lengths (8-64 bits), the stochastic implementations are over an order of magnitude more efficient than binary implementations of corresponding bit-widths (3-6 bits). However, as the computation precision is increased, the increase in energy is more pronounced for stochastic implementations compared to binary due to the exponential growth in bitstream length for a single bit increase in bit-width. This results in the stochastic implementations becoming energy inefficient for larger bitstream lengths. For CMOS stochastic implementation, the break-even point occurs when the bitstream length reaches 256 bits. In the case of SPINTASTIC, the intersection point is pushed further, by 4X, to bitstreams of length 1024 bits and to even higher bitstream lengths with vectorization. For almost all applications in the domains of image processing and RMS, in which stochastic computing has been widely explored, it has been demonstrated that bitstream length of 256 bits is sufficient to ensure correct functionality [135, 136, 139]. SPINTASTIC outperforms both binary and stochastic CMOS implementations in this regime by a significant margin. It is also noteworthy that SPINTASTIC is consistently superior to the stochastic CMOS baseline at all bitstream lengths.

8.6.3 Application-level analysis

The previous sections quantified the energy improvements of SPINTASTIC for datapath modules from various application domains. In this section, we study the benefits of SPINTASTIC at the application-level for three popular recognition applications *viz.* k-NN, GLVQ and SVM. Towards this end, we consider the stochastic recognition and mining processor proposed in [139] and replace its functional units with those designed using SPINTASTIC.

Figure 8.11 compares the normalized energy consumption of SPINTASTIC with the CMOS baseline implementations. We observe that, compared to binary CMOS,

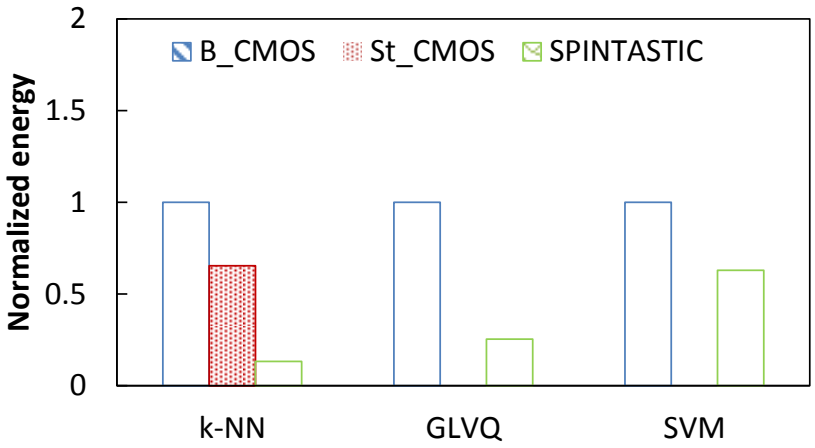


Fig. 8.11.: Application-level energy comparison

SPINTASTIC achieves 7.5X, 3.9X, and 1.6X reduction in energy for k-NN, GLVQ, and SVM, respectively. The higher improvement for k-NN stems from its low precision (5 bits) requirement [139], which enables stochastic computation with bitstreams of length 32. In contrast, SVM requires higher precision (8 bits), resulting in relatively small energy benefits. SPINTASTIC also outperforms stochastic CMOS implementations of these applications, with 4.9X, 3.5X and 2.4X improvement in energy for k-NN, GLVQ, and SVM, respectively.

8.7 Conclusion

In this work, we presented SPINTASTIC a new paradigm that uses stochastic computing to design logic with spintronic devices. We demonstrated the synergy between stochastic computing and spintronics and designed the different building blocks of stochastic logic with spintronic devices. Our experiments over a wide range of benchmarks circuits and popular applications demonstrate that SPINTASTIC is highly energy-efficient and is hence a promising direction to realize logic using spintronic devices.

9. DEVICE TO ARCHITECTURE SIMULATION FRAMEWORK

In order to evaluate STT-MRAM and DWM used in our designs, we have developed a device to architecture framework shown in Figure 9.1. At the device level, we have developed a self-consistent physics-based model to perform device simulations. At the circuit level, we have developed a CACTI-based cache simulation tool, Spin-CACTI, to compute some of the key metrics of STT-MRAM and DWM based caches. These cache characteristics are then used by the architectural simulators (described in Chapters 4-8) to compute the energy and performance at the system level. In this

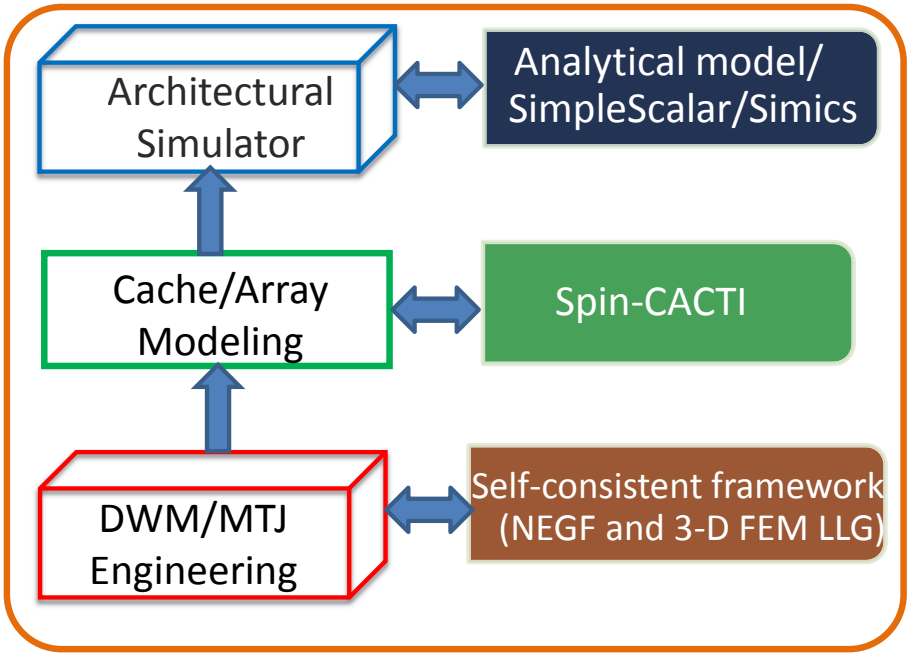


Fig. 9.1.: Device to architecture simulation framework

chapter, we present a detailed description of the methodology used to model STT-MRAM and DWM at device and circuit levels. This chapter is organized as follows:

We first describe the device-level modeling framework used to model MTJ and DWM in Section 9.1. We then describe the Spin-CACTI tool that can model STT-MRAM and DWM based cache in Section 9.2.

9.1 Spin device simulator

9.1.1 STT-MRAM

We model STT-MRAM device characteristics by self-consistently solving Non-Equilibrium Green's Function (NEGF) with Landau-Lifshitz-Gilbert (LLG) equation. In the NEGF approach, we first write the spin dependent Hamiltonian that gives the atomistic description of the MTJ structure subjected to an applied voltage. The Green's function is then calculated to determine the charge and spin currents through the MTJ. The torque on the free layer is calculated by taking the divergence of spin currents at the surfaces of the free layer. The torques from NEGF are used by LLG to simulate magnetization dynamics. The LLG is self-consistently coupled to the NEGF model through the free layer magnetization vector [146]. Memory arrays and bit-cells are simulated using the calculated I-V characteristics and spin-transfer torques from NEGF simulation.

9.1.2 DWM

In order to evaluate DWM, we have developed a self-consistent simulation framework for accurately capturing the domain wall motion under spin-torque. The framework consists of two self-consistent procedures: (a) spin-diffusion equation [142], and (b) a 2-D solution to the Landau-Lifshitz-Gilbert equation for capturing the dynamics of electron spins inside the entire nanomagnet. By self-consistently solving spin-diffusion equation and 2-D LLG, we can observe the propagation (movement) of the domain wall under an applied voltage. Note that in domain wall movement there is no physical movement of the wall; instead the spins inside the domain grad-

ually change their orientation when current is applied. We have benchmarked the simulation results using the proposed framework with experimental data in [147]. We extended the framework to include multiple domains within a single DWM tape. As a result, the accuracy of the simulations is ensured with the help of DW atomistic simulation capability we have developed. Based on the energy dissipation and latency of the domain wall motion, we computed the energy and latency of shift operations in the DWM tapes. The energy and latency to perform read/write operations at the read/write ports were computed in a manner similar to the read/write operation from STT-MRAM.

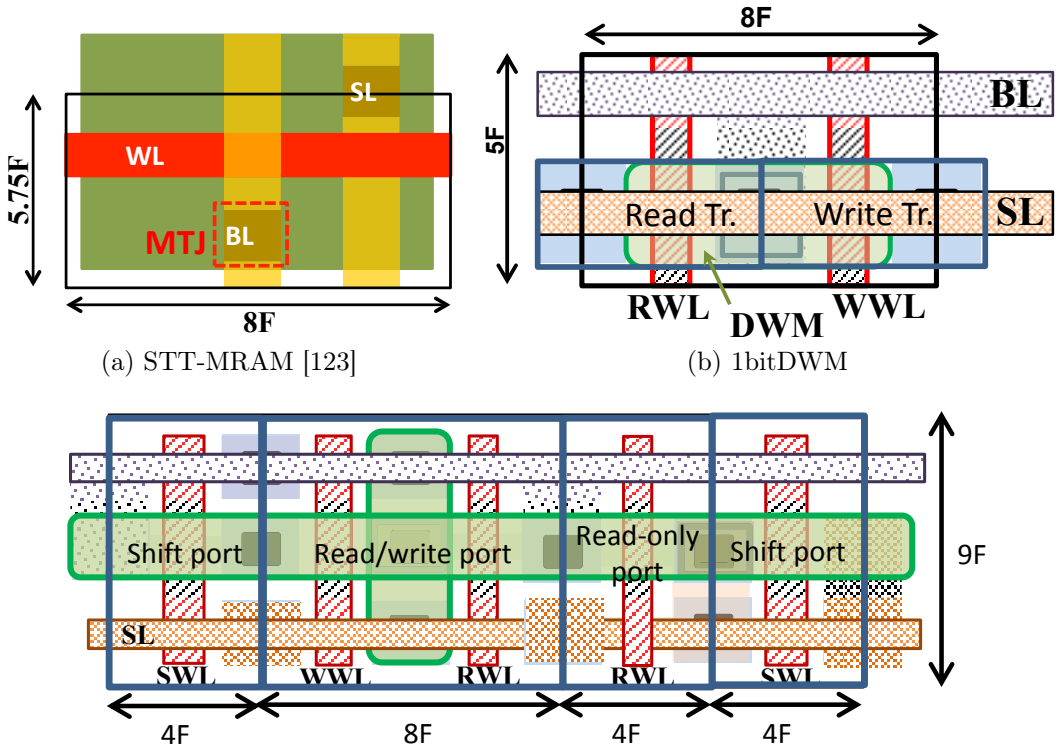
9.2 Spin-CACTI

In this section, we present Spin-CACTI, a CACTI-based cache simulator tool for evaluating spin-based caches. The key metrics that are of interest while evaluating a cache are its area, read/write energy, read/write latency and leakage power. In the case of DWM, we additionally require the shift energy/latency.

9.2.1 Area

The first step in the estimation of the cache area is to evaluate bit-cell area for different memory technologies. Figure 9.2 shows the bit-cell layout of different memory technologies. For estimating the area, we use the mosis layout rules and assume that each domain and the MTJ are sized $2F * F$, where F is the minimum feature size.

STT-MRAM bit-cell: Figure 9.2a shows the layout of STT-MRAM bit-cell consisting of an MTJ and an access transistor. The access transistor in the STT-MRAM bit-cell is sized large enough to supply the required write current. Further, we assume that the MTJ is integrated with CMOS using 3D technology and the area occupied by the bit-cell is dominated by the access transistor. Using the layout shown in Figure 9.2a, the area of STT-MRAM bit-cell is estimated to be $46F^2$.



(a) STT-MRAM [123]

(b) 1bitDWM

(c) MultibitDWM

Fig. 9.2.: Layout of STT-MRAM, 1bitDWM, and multibitDWM bit-cells

1bitDWM bit-cell: Figure 9.2b shows the layout of an 1bitDWM bit-cell used to estimate its area. Similar to STT-MRAM, we assume that ferromagnetic wire and the MTJ are integrated with the CMOS using 3D technology and the area occupied by the 1bitDWM bit-cell is dominated by the access transistors. Due to reduced current requirement for both read and write operations, the access transistors are minimum-sized and we estimate the area occupied by 1bitDWM bit-cell to be $40F^2$.

MultibitDWM bit-cell area: The area occupied by the multibitDWM bit-cell depends on the number of bits stored in the ferromagnetic wire, number of read/write ports, number of read-only ports and the number of extra bits introduced to avoid overflow due to shifts. Figure 9.2c shows an example layout of the multibitDWM bit-cell consisting of 1 read/write port, 1 read-only port and 2 shift ports. As shown in the figure, the width of the cell is given by $W = 9F$. The length of the multibitDWM bit-cell depends on the length of the ferromagnetic wire and the access transistors.

The length of the ferromagnetic wire is determined by sum of the number of bits stored in the multibitDWM bit-cell and the number of extra bits. The number of extra bits, in turn, is determined by the maximum number of shift operations that will be performed in either direction during read/write access. For a multibitDWM bit-cell that stores N_b bits per cell and has N_{rw} read/write ports, the number of extra bits can be estimated as N_b/N_{rw}. For instance, a multibitDWM bit-cell consisting of 16 bits with 2 read/write ports distributed uniformly will require a maximum of 4 shifts and therefore 8 extra bits to prevent loss of data. Therefore, the length of ferromagnetic wire is given by:

$$L_{fw} = (N_b + N_b/N_{rw}) * 2F \tag{9.1}$$

The length of the access transistors is determined number of read/write ports and read-only ports. In order to share the contacts across the access transistors and reduce the area overhead, we assume that the access ports are arranged in (S)(RW)(RW)..(RW)(S) pattern, where S indicates shift port, R indicates read-only ports and W indicates a read/write port. The length of the access transistors is given by

$$L_{at} = (8F * N_{rw}) + (4F * N_r) + 8F \tag{9.2}$$

Using equations 9.1 and 9.2, we compute the length of the multibitDWM bit-cell to be $L = Max(L_{fw}, L_{at})$ and the area of the multibitDWM bit-cell to be $Area = L * W$.

The area occupied by the cells is then used to compute the cache area and the capacitance of bitlines sourcelines and wordlines. These capacitances are then used to compute the energy and latency of the read and write operations inside a modified CACTI tool as described below.

9.2.2 Dynamic energy

In order to compute the energy consumed while performing the read/write operations, we first compute the different energy components like energy dissipated while charging/discharging of bitlines, sourcelines and wordlines, energy dissipated in MTJ during read/write operation, energy required to perform shift operations in the case of DWM, *etc.* For an array with N_{row} rows and N_{col} columns, the energy consumed in precharging bitline (BL) and sourceline (SL) to a voltage of V_{BL} and V_{SL}, respectively is given by:

$$E_{BL/SL} = N_{col}(C_{BL}V_{BL}^2 + C_{SL}V_{SL}^2) \tag{9.3}$$

where C_{BL} and C_{SL} are the capacitances of BL and SL, respectively. The exact voltage values are determined by the type of memory technology (STT-MRAM or DWM) and the desired operation (read/write/shift). For example, in order to write 0 to STT-MRAM, $V_{BL} = V_{write}$ and $V_{SL} = 0$ and the conditions are reversed for writing 1. The energy consumed in switching the wordline during read and write operation is given by:

$$E_{WL} = (C_{WL} + N_{row}C_{gate})V_{WL}^2 \tag{9.4}$$

where C_{WL} is the wordline capacitance and C_{gate} is the gate capacitance of the access transistor. Energy dissipated in the MTJ during read and write operation is computed using:

$$E_{MTJ} = |V_{BL-SL}|I_{MTJ}T_{active} \tag{9.5}$$

where V_{BL-SL} is the voltage applied across the BL and SL, I_{MTJ} is the current through the MTJ, and T_{active} is the time duration. In the case of write operations, T_{active} is the write pulse duration required to write into the cell. In the case of read operations, T_{active} is the time required to sense the current. I_{MTJ} is much higher for

write operations than read operations. In the case of DWM, the energy required to perform shift is computed as follows:

$$E_{shift} = N_{shift}|V_{BL-SL}|I_{shift}T_{shift} \tag{9.6}$$

where V_{BL-SL} is the net voltage applied across the BL and SL, which determines the current through the MTJs (I_{MTJ}) and the ferromagnetic wire (I_{shift}).

In addition to above mentioned components, peripheral circuits also contribute to the total energy consumption, which is modeled in a manner similar to conventional CACTI tool.

9.2.3 Leakage power

Another key metric of interest while evaluating a cache is its leakage power. Spin-based memories are non-volatile memories and the voltages of all the bitlines and wordlines can be set to 0V during idle state. Therefore, there is no leakage current within the bit-cell. The leakage energy of spin-based cache arises mainly from the peripheral circuitry. If the SRAM bit-cell is used to implement a part of the cache like tag array, then SRAM will also contribute to the leakage power consumption of the cache. During read and write operations, application of voltage across BL and SL leads to leakage current through all the unselected cells in the column, which also contributes to the total energy consumption.

9.2.4 Access latency

The read/write latency of spin-based memories are modeled by making suitable modifications to the CACTI tool. The read latency of a cache array depends on the time required to switch the wordline (T_{WL}), time required to precharge bitlines to appropriate voltages ($T_{precharge}$), time required to perform shift operations (T_{shift}), time required to sense the current through the MTJ (T_{cell}) and time required to drive

the data value to the output wires (T_{output}). Precharging bitlines occurs along with the decoding stage. The total read latency, therefore, is given by

$$T_{Read} = max(T_{WL}, T_{precharge}) + T_{cell} + T_{shift} + T_{output} \tag{9.7}$$

Similarly, the write latency of a cache array depends on the time required to switch the wordline (T_{WL}), time required to precharge the bitlines to appropriate voltages ($T_{precharge}$), time required to perform shift operation (T_{shift})and the write pulse duration (T_{write}). The write latency can be computed using equation 9.8.

$$T_{Write} = max(T_{WL}, T_{precharge}) + T_{shift} + T_{write} \tag{9.8}$$

For memories, that donot require any shift operations, $T_{shift} = 0$. The CACTI tool was suitably modified to model the read and write latencies in a DWM-based cache.

In this way, Spin-CACTI can be used to compute different key metrics, *viz.* area, read/write latencies, leakage power, and read/write energies, of STT-MRAM and DWM based cache designs.

10. CONCLUSIONS

Spintronic devices posses attractive characteristic like non-volatility, high integration density *etc.*, which make them promising candidates for future computing platforms. In this thesis, we presented a comprehensive circuit-to-architecture exploration for realizing both spin-based memory and logic. We showed that synergistically designing suitable circuits and architectures taking into consideration the unique characteristics of spintronic devices leads to significant benefits.

We designed a domain-specific many-core RM processor for an emerging class of data intensive applications– recognition, mining and synthesis. We showed that domain-specific architectures offer considerable flexibility to match the device characteristics with that of application and architecture. We proposed a heterogeneous memory hierarchy consisting of STT-MRAM and DWM and showed that the proposed architecture can maximally exploit the density and energy-efficiency of spintronic memories and significantly benefit highly parallel, data-intensive workloads such as recognition and mining.

We investigated the design of general-purpose cache hierarchy using spintronic memories. We proposed TapeCache and TAPESTRI for designing an energy-efficient all-spin cache based on DWM. TapeCache addresses one of the key challenges in the design of DWM-based cache – performance penalty arising from sequential access to data stored in DWM. In this work, we proposed suitable circuit-architecture co-design techniques that exploit the inherent read-write asymmetry and spatial locality of cache accesses to mask the increased latency from sequential accesses. In TAPESTRI, we addressed the other design challenge with the use of DWM – Inefficient write operation leading to high write energy/latency. We proposed different genres of DWM bit-cells that are optimized for performance/density. The proposed designs enable DWM to be used in all the levels in the cache hierarchy, including L1 cache,

where the spin-based memories have hitherto not been used due to their high write latency/energy. Our results showed that DWMs offer great potential in improving the energy-performance profile of general-purpose cache architectures.

We explored the use of spintronic memories in general-purpose graphics processing units (GPGPUs). We proposed STAG, a spintronic-tape architecture for GPGPU cache hierarchies. STAG utilizes different DWM bit-cells for designing different arrays in the on-chip memory hierarchy. We proposed suitable architectural techniques that exploits the unique characteristics of GPGPU architectures and applications to beneficially employ DWMs in GPGPUs. STAG achieves very high energy savings as well as performance improvements, thereby demonstrating the promise of DWM-based GPGPU memory hierarchy.

In the context of spintronic logic, we presented SPINTASTIC, a new design paradigm that uses stochastic computing to realize logic with spintronic devices. In SPINTASTIC, we established the synergy between stochastic computing and spintronics. We demonstrated the design of different building blocks of stochastic circuits by exploiting physical characteristics of spintronic devices. SPINTASTIC achieves significant improvements in energy compared to CMOS-based binary and stochastic implementations, thereby underscoring its efficiency.

In summary, spintronics is indeed a highly promising technology for the design of future computing platforms. Spintronic devices are especially suited for designing dense, energy-efficient memories. This thesis expands the use of spintronic memories to newer applications (*e.g.* DWM as on-chip caches), and proposes suitable circuit and architectural techniques that significantly enhances the benefits of spintronic memory designs. On the other hand, while spintronic logic faces significant challenges as drop-in replacement for CMOS, they are highly promising in application domains whose computational characteristics matches those of spintronic devices. This thesis presents one such direction for realizing logic with spintronic devices.

REFERENCES

REFERENCES

[1] A. C. Seabaugh and Q. Zhang, "Low-Voltage Tunnel Transistors for Beyond CMOS Logic," *Proceedings of the IEEE*, vol. 98, no. 12, pp. 2095–2110, Dec. 2010.

[2] X. Zhou, F. Hocke, A. Schliesser, A. Marx, H. Huebel, R. Cross, and T. Kippenberg, "Slowing, advancing and switching of microwave signals using circuit nanoelectromechanics," *Nature Physics*, vol. 9, pp. 179–184, Jan. 2013.

[3] P. Avouris, J. Appenzeller, R. Martel, and S. Wind, "Carbon nanotube electronics," *Proceedings of the IEEE*, vol. 91, no. 11, pp. 1772–1784, Nov. 2003.

[4] L. Liao, Y.-C. Lin, M. Bao, R. Cheng, J. Bai, Y. Liu, Y. Qu, K. L. Wang, Y. Huang, and X. Duan, "High-speed graphene transistors with a self-aligned nanowire gate," *Nature*, vol. 467, no. 7313, pp. 305–308, Sep. 2010.

[5] K. Bernstein, R. Cavin, W. Porod, A. Seabaugh, and J. Welser, "Device and Architecture Outlook for Beyond CMOS Switches," *Proceedings of the IEEE*, vol. 98, no. 12, pp. 2169 –2184, Dec. 2010.

[6] S. A. Wolf, J. Lu, M. Stan, E. Chen, and D. Treger, "The promise of nanomagnetics and spintronics for future logic and universal memory," *Proceedings of the IEEE*, vol. 98, no. 12, pp. 2155–2168, Dec. 2010.

[7] S. Sugahara and J. Nitta, "Spin-transistor electronics: An overview and outlook," *Proceedings of the IEEE*, vol. 98, no. 12, pp. 2124–2154, Dec. 2010.

[8] H. Iwai, "CMOS downsizing toward sub-10 nm," *Solid-State Electronics*, vol. 48, no. 4, pp. 497 – 503, Apr. 2004.

[9] M. Pinto, W. Brinkman, and W. Troutman, "The transistor's discovery and what's ahead," in *Proceeding of the European Solid-State Device Research Conference*, Sep. 1997, pp. 125 – 132.

[10] S. Tehrani, J. Slaughter, M. Deherrera, B. Engel, N. Rizzo, J. Salter, M. Durlam, R. Dave, J. Janesky, B. Butcher, K. Smith, and G. Grynkewich, "Magnetoresistive random access memory using magnetic tunnel junctions," *Proceedings of the IEEE*, vol. 91, no. 5, pp. 703 – 714, May 2003.

[11] A. Smith and Y. Huai, "STT-RAM - A New Spin on Universal Memory," *Future Fab Intl.*, vol. 23, Jul. 2007.

[12] K. Lee and S. Kang, "Development of Embedded STT-MRAM for Mobile System-on-Chips," *IEEE Transactions on Magnetics*, vol. 47, no. 1, pp. 131 –136, Jan. 2011.

[13] R. Desikan, C. Lefurgy, S. Keckler, and D. Burger, "On-chip MRAM as a High-Bandwidth, Low-Latency Replacement for DRAM Physical Memories," In IBM Austin CASC, Tech. Rep., Sep. 2002.

[14] S. Parkin, M. Hayashi, and L. Thomas, "Magnetic Domain-Wall Racetrack Memory," *Science*, vol. 320, no. 5873, pp. 190–194, Apr. 2008.

[15] R. Venkatesan, V. Kozhikkottu, C. Augustine, A. Raychowdhury, K. Roy, and A. Raghunathan, "TapeCache: A high density, energy efficient cache based on domain wall memory," in *Proceedings of the International Symposium on Low Power Electronics and Design*, Jul. 2012, pp. 185–190.

[16] R. Venkatesan, M. Sharad, K. Roy, and A. Raghunathan, "DWM-TAPESTRI - An energy efficient all-spin cache using domain wall shift based writes," in *Proceedings of the Design, Automation Test in Europe*, Apr. 2013, pp. 1825–1830.

[17] Z. Sun, W. Wu, and H. Li, "Cross-layer racetrack memory design for ultra high density and low power consumption," in *Proceedings of the Design Automation Conference*, Jun. 2013, pp. 53:1–53:6.

[18] T. Kawahara, R. Takemura, K. Miura, J. Hayakawa, S. Ikeda, Y. M. Lee, R. Sasaki, Y. Goto, K. Ito, T. Meguro, F. Matsukura, H. Takahashi, H. Matsuoka, and H. Ohno, "2 Mb SPRAM (SPin-Transfer Torque RAM) With Bit-by-Bit Bi-Directional Current Write and Parallelizing-Direction Current Read," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 109 –120, Jan. 2008.

[19] K. Miura, T. Kawahara, R. Takemura, J. Hayakawa, S. Ikeda, R. Sasaki, H. Takahashi, H. Matsuoka, and H. Ohno, "A novel SPRAM (SPin-transfer torque RAM) with a synthetic ferrimagnetic free layer for higher immunity to read disturbance and reducing write-current dispersion," in *Proceedings of the IEEE Symposium on VLSI Technology*, Jun. 2007, pp. 234 –235.

[20] A. Annunziata, M. Gaidis, L. Thomas, C. Chien, C. Hung, P. Chevalier, E. O'Sullivan, J. Hummel, E. Joseph, Y. Zhu, T. Topuria, E. Delenia, P. Rice, S. Parkin, and W. Gallagher, "Racetrack memory cell array with integrated magnetic tunnel junction readout," in *Proceedings of the International Electron Devices Meeting*, Dec. 2011, pp. 24.3.1 –24.3.4.

[21] S. Fukami, T. Suzuki, K. Nagahara, N. Ohshima, Y. Ozaki, S. Saito, R. Nebashi, N. Sakimura, H. Honjo, K. Mori, C. Igarashi, S. Miura, N. Ishiwata, and T. Sugibayashi, "Low-Current Perpendicular Domain Wall Motion Cell for Scalable High-Speed MRAM," in *Proceedings of the IEEE Symposium on VLSI Technology*, Jun. 2009, pp. 230 –231.

[22] Everspin Technologies, "www.everspin.com."

[23] D. Bhowmik, L. You, and S. Salahuddin, "Spin hall effect clocking of nanomagnetic logic without a magnetic field," *Nature Nanotechnology*, vol. 9, no. 1, pp. 59–63, Jan. 2014.

[24] B. Behin-Aein, D. Datta, S. Salahuddin, and S. Datta, "Proposal for an all-spin logic device with built-in memory," *Nature Nanotechnology*, vol. 5, no. 4, pp. 266–270, Feb. 2010.

[25] A. Imre, L. Csaba, G. andJi, A. Orlov, G. H. Bernstein, and W. Porod, "Majority logic gate for magnetic quantum-dot cellular automata," *Science*, vol. 311, no. 5758, pp. 205–208, Jan. 2006.

[26] D.A.Allwood, G.Xiong, C.C.Faulkner, D. Atkinson, D. Petit, and R. Cowburn, "Magnetic Domain-Wall Logic," *Science*, vol. 309, no. 5741, pp. 1688–1692, Sep. 2005.

[27] J. Currivan, Y. Jang, M. Mascaro, M. Baldo, and C. Ross, "Low Energy Magnetic Domain Wall Logic in Short, Narrow, Ferromagnetic Wires," *IEEE Magnetics Letters*, vol. 3, p. 3000104, Apr. 2012.

[28] D. Morris, D. Bromberg, J.-G. J. Zhu, and L. Pileggi, "mLogic: ultra-low voltage non-volatile logic circuits using STT-MTJ devices," in *Proceedings of the Design Automation Conference*, Jun. 2012, pp. 486–491.

[29] M. Sharad, C. Augustine, G. Panagopoulos, and K. Roy, "Spin-Based Neuron Model With Domain-Wall Magnets as Synapse," *IEEE Transactions on Nanotechnology*, vol. 11, no. 4, pp. 843 –853, Jul. 2012.

[30] Intel Corporation, "www.intel.com."

[31] C. Kenyon, A. Kornfeld, K. Kuhn, M. Liu, A. Maheshwari, W. Shih, S. Sivakumar, G. Taylor, P. VanDerVoorn, and K. Zawadzki, "Managing Process Variation in Intel's 45nm CMOS Technology," *Intel Technology Journal*, Jun. 2008.

[32] K. Bernstein, D. J. Frank, A. E. Gattiker, W. Haensch, B. L. Ji, S. R. Nassif, E. J. Nowak, D. J. Pearson, and N. J. Rohrer, "High-performance CMOS variability in the 65-nm regime and beyond," *IBM Journal of Research and Development*, vol. 50, no. 4.5, pp. 433 –449, Jul. 2006.

[33] H. Sutter, "The free lunch is over: A fundamental turn toward concurrency in software," *Dr. Dobb's journal*, vol. 30, no. 3, pp. 497 – 503, Mar. 2005.

[34] B. Meyerson, "Opening Keynote Address - How does one define "technology" now that classical scaling is dead (and has been for years)?" in *Proceedings of the Design Automation Conference*, Jun. 2005.

[35] J. Stathis and S. Zafar, "The negative bias temperature instability in MOS devices: A review," *Microelectronics Reliability*, vol. 46, no. 2-4, pp. 270–286, Feb.-Apr. 2006.

[36] T. M. Mak, "Is CMOS more reliable with scaling," in *CRC-IEEE Bast Workshop*, Feb. 2003.

[37] [Online]. Available: www.karlrupp.net/2013/06/cpu-gpu-and-mic-hardware-characteristics-over-time

[38] Nvidia, "www.nvidia.com."

[39] AMD, "www.amd.com."

[40] Y. Chen, J. Chhugani, P. Dubey, C. Hughes, D. Kim, S. Kumar, V. Lee, A. Nguyen, and M. Smelyanskiy, "Convergence of Recognition, Mining, and Synthesis Workloads and Its Implications," *Proceedings of the IEEE*, vol. 96, no. 5, pp. 790 –807, May 2008.

[41] X. Wu, J. Li, L. Zhang, E. Speight, R. Rajamony, and Y. Xie, "Design exploration of hybrid caches with disparate memory technologies," *ACM Transactions on Architecture and Code Optimization*, vol. 7, no. 3, pp. 15:1–15:34, Dec. 2010.

[42] A. Nigam, C. W. Smullen, IV, V. Mohan, E. Chen, S. Gurumurthi, and M. R. Stan, "Delivering on the promise of universal memory for Spin-Transfer Torque RAM (STT-RAM)," in *Proceedings of the International symposium on Low-power electronics and design*, Aug. 2011, pp. 121–126.

[43] Y. Xie, "Modeling, Architecture, and Applications for Emerging Memory Technologies," *IEEE Design and Test of Computers*, vol. 28, no. 1, pp. 44 –51, Jan.-Feb. 2011.

[44] M. K. Qureshi, V. Srinivasan, and J. A. Rivers, "Scalable high performance main memory system using phase-change memory technology," in *Proceedings of the international symposium on Computer architecture*, Jun. 2009, pp. 24–33.

[45] G. Dhiman, R. Ayoub, and T. Rosing, "PDRAM: a hybrid PRAM and DRAM main memory system," in *Proceedings of the Design Automation Conference*, Jun. 2009, pp. 664–469.

[46] H. Wong, S. Raoux, S. Kim, J. Liang, J. Reifenberg, B. Rajendran, M. Asheghi, and K. Goodson, "Phase Change Memory," *Proceedings of the IEEE*, vol. 98, no. 12, pp. 2201 –2227, Dec. 2010.

[47] R. Venkatesan, V. Chippa, C. Augustine, K. Roy, and A. Raghunathan, "Energy efficient many-core processor for recognition and mining using spin-based memory," in *Proceedings of the IEEE/ACM International Symposium on Nanoscale Architectures*, Jun. 2011, pp. 122 –128.

[48] H. Manem, G. Rose, X. He, and W. Wang, "Design considerations for variation tolerant multilevel CMOS/Nano memristor memory," in *Proceedings of the Great Lakes Symposium on VLSI*, May 2010, pp. 287–292.

[49] D. B. Strukov, D. R. Stewart, J. Borghetti, X. Li, M. Pickett, G. Medeiros-Ribeiro, W. Robinett, G. S. Snider, J. P. Strachan, W. Wu, Q. Xia, J. J. Yang, and R. S. Williams, "Hybrid CMOS/memristor circuits," in *Proceedings ISCAS*, May 2010, pp. 1967–1970.

[50] C. Xu, X. Dong, N. P. Jouppi, and Y. Xie, "Design implications of memristor-based RRAM cross-point structures," in *Proceedings of the Design, Automation Test in Europe*, Mar. 2011, pp. 1 –6.

[51] R. Venkatesan, V. Chippa, C. Augustine, K. Roy, and A. Raghunathan, "Domain-specific many-core computing using spin-based memory," *IEEE Transactions on Nanotechnology*, vol. 13, no. 5, pp. 881–894, Sept 2014.

[52] N. Mojumder, S. Gupta, S. Choday, D. Nikonov, and K. Roy, "A Three-Terminal Dual-Pillar STT-MRAM for High-Performance Robust Memory Applications," *IEEE Transactions on Electron Devices*, vol. 58, no. 5, pp. 1508 –1516, May 2011.

[53] N. Mojumder and K. Roy, "Proposal for switching current reduction using reference layer with tilted magnetic anisotropy in magnetic tunnel junctions for spin-transfer torque (stt) mram," *IEEE Transactions on Electron Devices*, vol. 59, no. 11, pp. 3054–3060, Nov. 2012.

[54] C. Augustine, A. Raychowdhury, D. Somasekhar, J. Tschanz, K. Roy, and V. De, "Numerical analysis of typical STT-MTJ stacks for 1T-1R memory arrays," in *Proceedings of the International Electron Devices Meeting*, Dec. 2010, pp. 22.7.1 –22.7.4.

[55] Z. Diao, Z. Li, S. Wang, Y. Ding, A. Panchula, E. Chen, L.-C. Wang, and Y. Huai, "Spin-transfer torque switching in magnetic tunnel junctions and spin-transfer torque random access memory," *Journal of Physics: Condensed Matter*, vol. 19, no. 16, p. 165209, Apr. 2007.

[56] W. Zhao, J. Duval, J.-O. Klein, and C. Chappert, "A compact model for magnetic tunnel junction (MTJ) switched by thermally assisted Spin transfer torque (TAS + STT)," *Nanoscale Research Letters*, vol. 6, no. 1, p. 368, Apr. 2011.

[57] M. Carpentieri, M. Ricci, P. Burrascano, L. Torres, and G. Finocchio, "Wideband microwave signal to trigger fast switching processes in magnetic tunnel junctions," *Journal of Applied Physics*, vol. 111, no. 7, p. 07C909, Feb. 2012.

[58] G. E. Rowlands, T. Rahman, J. A. Katine, J. Langer, A. Lyle, H. Zhao, J. G. Alzate, A. A. Kovalev, Y. Tserkovnyak, Z. M. Zeng, H. W. Jiang, K. Galatsis, Y. M. Huai, P. K. Amiri, K. L. Wang, I. N. Krivorotov, and J.-P. Wang, "Deep subnanosecond spin torque switching in magnetic tunnel junctions with combined in-plane and perpendicular polarizers," *Applied Physics Letters*, vol. 98, no. 10, p. 102509, Mar. 2011.

[59] M. Carpentieri, M. Ricci, P. Burrascano, L. Torres, and G. Finocchio, "Noise-Like Sequences to Resonant Excite the Writing of a Universal Memory Based on Spin-Transfer-Torque MRAM," *IEEE Transactions on Magnetics*, vol. 48, no. 9, pp. 2407–2414, Sep. 2012.

[60] J. Li, H. Liu, S. Salahuddin, and K. Roy, "Variation-tolerant Spin-Torque Transfer (STT) MRAM array for yield enhancement," in *Proceedings Custom Integrated Circuits Conference*, Sep. 2008, pp. 193 –196.

[61] D. Lee, S. K. Gupta, and K. Roy, "High-performance low-energy STT MRAM based on balanced write scheme," in *Proceedings of the International Symposium on Low Power Electronics and Design*, Jul. 2012, pp. 9–14.

[62] Y. Kim, S. K. Gupta, S. P. Park, G. Panagopoulos, and K. Roy, "Write-optimized reliable design of STT MRAM," in *Proceedings of the International Symposium on Low Power Electronics and Design*, Jul. 2012, pp. 3–8.

[63] P. Zhou, B. Zhao, J. Yang, and Y. Zhang, "Energy reduction for STT-RAM using early write termination," in *Proceedings of the International Conference on Computer-Aided Design*, Nov. 2009, pp. 264 –268.

[64] J. Li, C. Augustine, S. Salahuddin, and K. Roy, "Modeling of failure probability and statistical design of Spin-Torque Transfer Magnetic Random Access Memory (STT MRAM) array for yield enhancement," in *Proceedings of the Design Automation Conference*, Jun. 2008, pp. 278 –283.

[65] Y. Chen and H. Li, "Emerging sensing techniques for emerging memories," in *Proceedings of the Asia and South Pacific Design Automation Conference*, Jan. 2011, pp. 204 –210.

[66] Y. Chen, W. F. Wong, H. Li, and C. K. Koh, "Processor caches with multi-level spin-transfer torque RAM cells," in *Proceedings of the International Symposium on Low Power Electronics and Design*, Aug. 2011, pp. 73–78.

[67] S. Chatterjee, M. Rasquinha, S. Yalamanchili, and S. Mukhopadhyay, "A methodology for robust, energy efficient design of Spin-Torque-Transfer RAM arrays at scaled technologies," in *Proceedings of the International Conference on Computer-Aided Design*, Nov. 2009, pp. 474 –477.

[68] F. Pellizzer, A. Pirovano, F. Ottogalli, M. Magistretti, M. Scaravaggi, P. Zuliani, M. Tosi, A. Benvenuti, P. Besana, S. Cadeo, T. Marangon, R. Morandi, R. Piva, A. Spandre, R. Zonca, A. Modelli, E. Varesi, T. Lowrey, A. Lacaita, G. Casagrande, P. Cappelletti, and R. Bez, "Novel u-trench phase-change memory cell for embedded and stand-alone non-volatile memory applications," in *Proceedings of the IEEE Symposium on VLSI Technology*, Jun. 2004, pp. 18 – 19.

[69] F. Pellizzer, A. Benvenuti, B. Gleixner, Y. Kim, B. Johnson, M. Magistretti, T. Marangon, A. Pirovano, R. Bez, and G. Atwood, "A 90nm Phase Change Memory Technology for Stand-Alone Non-Volatile Memory Applications," in *Proceedings of the IEEE Symposium on VLSI Technology*, Jun. 2006, pp. 122 –123.

[70] A. Pirovano, F. Pellizzer, I. Tortorelli, R. Harrigan, M. Magistretti, P. Petruzza, E. Varesi, D. Erbetta, T. Marangon, F. Bedeschi, R. Fackenthal, G. Atwood, and R. Bez, "Self-aligned u-trench phase-change memory cell architecture for 90nm technology and beyond," in *Proceedings of the European Solid State Device Research Conference*, Sep. 2007, pp. 222 –225.

[71] G. Servalli, "A 45nm generation Phase Change Memory technology," in *Proceedings of the International Electron Devices Meeting*, Dec. 2009, pp. 1 –4.

[72] W. Chen, C. Lee, D. Chao, Y. Chen, F. Chen, C. Chen, R. Yen, M. Chen, W. Wang, T. Hsiao, J. Yeh, S. Chiou, M. Liu, T. Wang, L. Chein, C. Huang, N. Shih, L. Tu, D. Huang, T. Yu, M. Kao, and M.-J. Tsai, "A Novel Cross-Spacer Phase Change Memory with Ultra-Small Lithography Independent Contact Area," in *Proceedings of the International Electron Devices Meeting*, Dec. 2007, pp. 319 –322.

[73] Y. Ha, J. Yi, H. Horii, J. Park, S. Joo, S. Park, U.-I. Chung, and J. Moon, "An edge contact type cell for Phase Change RAM featuring very low power consumption," in *Proceedings of the IEEE Symposium on VLSI Technology*, Jun. 2003, pp. 175 – 176.

[74] C.-F. Chen, A. Schrott, M. Lee, S. Raoux, Y. Shih, M. Breitwisch, F. Baumann, E. Lai, T. Shaw, P. Flaitz, R. Cheek, E. Joseph, S. Chen, B. Rajendran, H. Lung, and C. Lam, "Endurance Improvement of Ge2Sb2Te5-Based Phase Change Memory," in *IEEE International Memory Workshop*, May 2009, pp. 1 –2.

[75] L. Jiang, Y. Zhang, and J. Yang, "Enhancing phase change memory lifetime through fine-grained current regulation and voltage upscaling," in *Proceedings of the International Symposium on Low Power Electronics and Design*, Aug. 2011, pp. 127 –132.

[76] S. S. P. Parkin, "Shiftable magnetic shift register and method of using the same," *U.S.Patent 6834005*, issued April 26, 2004.

[77] Positioning Bits in Nanowire Memory, "Tech. Review," Jan. 2011.

[78] E. R. Lewis, D. Petit, L. OâĂŹBrien, A. Fernandez-Pacheco, J. Sampaio, A.-V. Jausovec, H. T. Zeng, D. E. Read, and R. P. Cowburn, "Fast domain wall motion in magnetic comb structures," *Nature*, vol. 9, no. 12, pp. 980–983, Dec. 2010.

[79] L. Thomas, R. Moriya, C. Rettner, and S. Parkin, "Dynamics of Magnetic Domain Walls Under Their Own Inertia," *Science*, vol. 330, no. 6012, pp. 1810–1813, Dec. 2010.

[80] D. Chiba, G. Yamada, T. Koyama, K. Ueda, H. Tanigawa, S. Fukami, T. Suzuki, N. Ohshima, N. Ishiwata, Y. Nakatani, and T. Ono, "Control of Multiple Magnetic Domain Walls by Current in a Co/Ni Nano-Wire," *Applied Physics Express*, vol. 3, no. 073004, pp. 1 –3, Jul. 2010.

[81] W. Zhao, D. Ravelosona, J. Klein, and C. Chappert, "Domain Wall Shift Register-Based Reconfigurable Logic," *IEEE Transactions on Magnetics*, vol. 47, no. 10, pp. 2966 –2969, Oct. 2011.

[82] N. Goswami, B. Cao, and T. Li, "Power-performance co-optimization of throughput core architecture using resistive memory," in *Proceedings of the International Symposium on High Performance Computer Architecture*, Feb. 2013, pp. 342–353.

[83] X. Wu, J. Li, L. Zhang, E. Speight, R. Rajamony, and Y. Xie, "Hybrid cache architecture with disparate memory technologies," in *Proceedings of the International Symposium on Computer Architecture*, Jun. 2009, pp. 34–45.

[84] J. Hu, C. Xue, Q. Z., W. C. Tseng, and E. Sha, "Towards energy efficient hybrid on-chip scratch pad memory with non-volatile memory," in *Proceedings of the Design, Automation Test in Europe*, Mar. 2011, pp. 1 –6.

[85] J. Cong, K. Gururaj, H. Huang, C. Liu, G. Reinman, and Y. Zou, "An energy-efficient adaptive hybrid cache," in *Proceedings of the International Symposium on Low Power Electronics and Design*, Aug. 2011, pp. 67–72.

[86] J. Zhao and Y. Xie, "Optimizing bandwidth and power of graphics memory with hybrid memory technologies and adaptive data migration," in *Proceedings of the International Conference on Computer-Aided Design*, Nov. 2012, pp. 81–87.

[87] B. Wang, B. Wu, D. Li, X. Shen, W. Yu, Y. Jiao, and J. Vetter, "Exploring hybrid memory for GPU energy efficiency through software-hardware co-design," in *Proceedings of the International Conference on Parallel Architectures and Compilation Techniques*, Sep. 2013, pp. 93–102.

[88] X. Wu, J. Li, L. Zhang, E. Speight, and Y. Xie, "Power and performance of read-write aware Hybrid Caches with non-volatile memories," in *Proceedings of the Design, Automation Test in Europe*, Apr. 2009, pp. 737 –742.

[89] A. Jadidi, M. Arjomand, and H. S. Azad, "High-endurance and performance-efficient design of hybrid cache architectures through adaptive line replacement," in *Proceedings of the International Symposium on Low Power Electronics and Design*, Aug. 2011, pp. 79–84.

[90] M. Rasquinha, D. Choudhary, S. Chatterjee, S. Mukhopadhyay, and S. Yalamanchili, "An energy efficient cache design using Spin Torque Transfer (STT) RAM," in *Proceedings of the International Symposium on Low Power Electronics and Design*, Aug. 2010, pp. 389 –394.

[91] P. Zhou, B. Zhao, J. Yang, and Y. Zhang, "A durable and energy efficient main memory using phase change memory technology," in *Proceedings of the International Symposium on Computer Architecture*, Jun. 2009.

[92] S. P. Park, S. Gupta, N. Mojumder, A. Raghunathan, and K. Roy, "Future cache design using STT MRAMs for improved energy efficiency: devices, circuits and architecture," in *Proceedings of the Design Automation Conference*, Jun. 2012, pp. 492–497.

[93] B. C. Lee, E. Ipek, O. Mutlu, and D. Burger, "Architecting phase change memory As a scalable DRAM alternative," in *Proceedings of the International Symposium on Computer Architecture*, Jun. 2009, pp. 2–13.

[94] G. Sun, X. Dong, Y. Xie, J. Li, and Y. Chen, "A novel architecture of the 3D stacked MRAM L2 cache for CMPs," in *Proceedings of the International Symposium on High Performance Computer Architecture*, Feb. 2009, pp. 239 –249.

[95] A. Mishra, X. Dong, G. Sun, Y. Xie, N. Vijaykrishnan, and C. Das, "Architecting on-chip interconnects for stacked 3D STT-RAM caches in CMPs," in *Proceedings of the International Symposium on Computer Architecture*, Jun. 2011, pp. 69–80.

[96] C. Smullen, V. Mohan, A. Nigam, S. Gurumurthi, and M. Stan, "Relaxing non-volatility for fast and energy-efficient STT-RAM caches," in *Proceedings of the International Symposium on High Performance Computer Architecture*, Feb. 2011, pp. 50 –61.

[97] A. Jog, A. Mishra, C. Xu, Y. Xie, V. Narayanan, R. Iyer, and C. Das, "Cache revive: Architecting volatile STT-RAM caches for enhanced performance in CMPs," in *Proceedings of the Design Automation Conference*, Jun. 2012, pp. 243 –252.

[98] S. Cho and H. Lee, "Flip-N-Write: A simple deterministic technique to improve PRAM write performance, energy and endurance," in *Microarchitecture, 2009. MICRO-42. 42nd Annual IEEE/ACM International Symposium on*, Dec. 2009, pp. 347 –357.

[99] G. Sun, D. Niu, J. Ouyang, and Y. Xie, "A frequent-value based PRAM memory architecture," in *Proceedings of the Asia and South Pacific Design Automation Conference*, Jan. 2011, pp. 211 –216.

[100] D. Kim, Y. Lee, J. Cai, I. Lauer, L. Chang, S. Koester, D. Sylvester, and D. Blaauw, "Low power circuit design based on heterojunction tunneling transistors (HETTs)," in *Proceedings of the International Symposium on Low Power Electronics and Design*, Aug. 2009, pp. 219–224.

[101] X. Yang and K. Mohanram, "Robust 6T Si tunneling transistor SRAM design," in *Proceedings of the Design, Automation Test in Europe*, Mar. 2011, pp. 1 –6.

[102] J. Singh, K. Ramakrishnan, S. Mookerjea, S. Datta, N. Vijaykrishnan, and D. Pradhan, "A novel Si-Tunnel FET based SRAM design for ultra low-power 0.3V VDD applications," in *Proceedings of the Asia and South Pacific Design Automation Conference*, Jan. 2010, pp. 181 –186.

[103] G. Csaba, A. Imre, G. Bernstein, W. Porod, and V. Metlushko, "Nanocomputing by field-coupled nanomagnets," *IEEE Transactions on Nanotechnology*, vol. 1, no. 4, pp. 209–213, Dec. 2002.

[104] C. Augustine, G. Panagopoulos, B. Behin-Aein, S. Srinivasan, A. Sarkar, and K. Roy, "Low-power functionality enhanced computation architecture using spin-based devices," in *Proceedings of the IEEE/ACM International Symposium on Nanoscale Architectures*, Jun. 2011, pp. 129–136.

[105] G. Snider, R. Amerson, D. Carter, H. Abdalla, M. Qureshi, J. LeÌĄ andveilleÌĄ and, M. Versace, H. Ames, S. Patrick, B. Chandler, A. Gorchetchnikov, and E. Mingolla, "From Synapses to Circuitry: Using Memristive Memory to Explore the Electronic Brain," *Computer*, vol. 44, no. 2, pp. 21 –28, Feb. 2011.

[106] L. Britnell, R. V. Gorbachev, R. Jalil, B. D. Belle, F. Schedin, A. Mishchenko, T. Georgiou, M. I. Katsnelson, L. Eaves, S. V. Morozov, N. M. R. Peres, J. Leist, A. K. Geim, K. S. Novoselov, and L. A. Ponomarenko, "Field-effect tunneling transistor based on vertical graphene heterostructures," *Science*, vol. 335, no. 6071, pp. 947–950, Feb. 2012.

[107] T. Krishnamohan, D. Kim, S. Raghunathan, and K. Saraswat, "Double-Gate Strained-Ge Heterostructure Tunneling FET (TFET) With record high drive currents and 60mV/dec subthreshold slope," in *Proceedings of the International Electron Devices Meeting*, Dec. 2008, pp. 1 –3.

[108] J. Smith, S. Das, and J. Appenzeller, "Broken-Gap Tunnel MOSFET: A Constant-Slope Sub-60-mV/decade Transistor," *IEEE Electron Device Letters*, vol. 32, no. 10, pp. 1367 –1369, Oct. 2011.

[109] D. Mohata, R. Bijesh, V. Saripalli, T. Mayer, and S. Datta, "Self-aligned gate nanopillar In0.53Ga0.47As vertical tunnel transistor," in *Proceedings of the Device Research Conference*, Jun. 2011, pp. 203 –204.

[110] S. Mookerjea, D. Mohata, R. Krishnan, J. Singh, A. Vallett, A. Ali, T. Mayer, V. Narayanan, D. Schlom, A. Liu, and S. Datta, "Experimental demonstration of 100nm channel length In0.53Ga0.47As-based vertical inter-band tunnel field effect transistors (TFETs) for ultra low-power logic and SRAM applications," in *Proceedings of the International Electron Devices Meeting*, Dec. 2009, pp. 1 –3.

[111] V. Saripalli, G. Sun, A. Mishra, Y. Xie, S. Datta, and V. Narayanan, "Exploiting Heterogeneity for Energy Efficiency in Chip Multiprocessors," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 1, no. 2, pp. 109 –119, Jun. 2011.

[112] V. Saripalli, A. Mishra, S. Datta, and V. Narayanan, "An energy-efficient heterogeneous CMP based on hybrid TFET-CMOS cores," in *Proceedings of the Design Automation Conference*, Jun. 2011, pp. 729 –734.

[113] M. Sharad, K. Yogendra, K.-W. Kwon, and K. Roy, "Design of ultra high density and low power computational blocks using nano-magnets," in *Proceedings of the International Symposium on Quality Electronic Design*, Mar. 2013, pp. 223–230.

[114] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The PARSEC benchmark suite: Characterization and architectural implications," in *Proceedings of the International Conference on Parallel Architectures and Compilation Techniques*, Oct. 2008, pp. 72–81.

[115] Design Compiler, "Synopsys inc."

[116] CACTI, "http://www.hpl.hp.com/research/cacti/."

[117] Nanosim, "Synopsys Inc."

[118] Y. LeCun and C. Cortes, "The MNIST database of handwritten digits," *http://yann.lecun.com/exdb/mnist/*.

[119] A. Frank and A. Asuncion, "UCI Machine Learning repositry," 2010.

[120] V. Vapnik, *The nature of statistical learning theory*. Springer-Verlag New York, Inc., 1995.

[121] J. MacQueen, "Some methods for classification and analysis of multivariate observation," in *Proceedings Berkeley Symposium on Mathematical Statistics and Probability*, 1967.

[122] N. Pal, J. Bezdek, and E.-K. Tsao, "Generalized clustering networks and Kohonen's self-organizing scheme," *IEEE Transactions on Neural Networks*, vol. 4, no. 4, pp. 549 –557, Jul. 1993.

[123] S. Gupta, S. P. Park, N. Mojumder, and K. Roy, "Layout-aware optimization of stt mrams," in *Proceedings of the Design, Automation Test in Europe*, Mar. 2012, pp. 1455 –1458.

[124] A. A. Khan, J. Schmalhorst, A. Thomas, O. Schebaum, and G. Reiss, "Dielectric breakdown in CoFeB/MgO/CoFeB magnetic tunnel junction," *Journal of Applied Physics*, vol. 103, pp. 123 705–123 705–5, Jun. 2008.

[125] T.-F. Chen and J.-L. Baer, "Effective hardware-based data prefetching for high-performance processors," *IEEE Transactions on Computers*, vol. 44, no. 5, pp. 609–623, May 1995.

[126] C. Augustine, A. Raychowdhury, B. Behin-Aein, S. Srinivasan, J. Tschanz, V. De, and K. Roy, "Numerical analysis of domain wall propagation for dense memory arrays," in *Proceedings of the International Electron Devices Meeting*, Dec. 2011, pp. 17.6.1 –17.6.4.

[127] T. Austin, E. Larson, and D. Ernst, "SimpleScalar: An Infrastructure for Computer System Modeling," *Computer*, vol. 35, pp. 59–67, Feb. 2002.

[128] W. Jia, K. A. Shaw, and M. Martonosi, "Characterizing and improving the use of demand-fetched caches in GPUs," in *Proceedings on International Conference on Supercomputing*, Jun. 2012, pp. 15–24.

[129] A. Bakhoda, G. Yuan, W. Fung, H. Wong, and T. Aamodt, "Analyzing cuda workloads using a detailed gpu simulator," in *IEEE International Symposium on Performance Analysis of Systems and Software*, Apr. 2009, pp. 163–174.

[130] S. Che, M. Boyer, J. Meng, D. Tarjan, J. Sheaffer, S.-H. Lee, and K. Skadron, "Rodinia: A benchmark suite for heterogeneous computing," in *Proceeding of the International Symposium on Workload Characterization*, Oct. 2009, pp. 44–54.

[131] J. A. Stratton, C. Rodrigues, I.-J. Sung, N. Obeid, vLi Wen Chang, N. Anssari, G. D. Liu, and W. mei W. Hwu, "Parboil: A revised benchmark suite for scientific and commercial throughput computing," in *IMPACT Technical Report*, Mar. 2012.

[132] M. T. Niemier, G. H. Bernstein, G. Csaba, A. Dingler, X. S. Hu, S. Kurtz, S. Liu, J. Nahas, W. Porod, M. Siddiq, and E. Varga, "Nanomagnet logic: progress toward system-level integration," *Journal of Physics: Condensed Matter*, vol. 23, no. 49, p. 493202, Dec. 2011.

[133] V. Calayir, D. Nikonov, S. Manipatruni, and I. Young, "Static and clocked spintronic circuit design and simulation with performance analysis relative to cmos," *IEEE Transactions on Circuits and Systems I*, vol. 61, no. 2, pp. 393–406, Feb. 2014.

[134] B. Gaines, "Stochastic computing systems," in *Advances in Information Systems Science*. Springer US, 1969, pp. 37–172.

[135] W. Qian, X. Li, M. D. Riedel, K. Bazargan, and D. Lilja, "An architecture for fault-tolerant computation with stochastic logic," *IEEE Transactions on Computers*, vol. 60, no. 1, pp. 93–105, Jan. 2011.

[136] A. Alaghi and J. Hayes, "Survey of stochastic computing," *ACM Transactions on Embedded Computing*, vol. 12, no. 2s, pp. 92:1–92:19, May 2013.

[137] B. Brown and H. Card, "Stochastic neural computation. i. computational elements," *IEEE Transactions Computers*, vol. 50, no. 9, pp. 891–905, Sep. 2001.

[138] S. Tehrani, S. Mannor, and W. Gross, "Fully parallel stochastic ldpc decoders," *IEEE Transactions on Signal Processing*, vol. 56, no. 11, pp. 5692–5703, Nov. 2008.

[139] V. K. Chippa, S. Venkataramani, K. Roy, and A. Raghunathan, "Storm: A stochastic recognition and mining processor," in *Proceedings of the International Symposium on Low Power Electronic Design*, Aug 2014, pp. 39–44.

[140] X. Fong, M. C. Chen, and K. Roy, "Generating true random numbers using on-chip complementary polarizer spin-transfer torque magnetic tunnel junctions," in *Proceedings of the Device Research Conference*, Jun. 2014, pp. 103–104.

[141] L. He, W. Doyle, and H. Fujiwara, "High speed coherent switching below the Stoner-Wohlfarth limit," *IEEE Transactions on Magnetics*, vol. 30, no. 6, pp. 4086–4088, Nov. 1994.

[142] T. Valet and A. Fert, "Theory of the perpendicular magnetoresistance in magnetic multilayers," *Physical Review B*, vol. 48, no. 10, pp. 7099–7113, Sep. 1993.

[143] B. Behin-Aein, A. Sarkar, S. Srinivasan, and S. Datta, "Switching energy-delay of all spin logic devices," *Applied Physics Letters*, vol. 98, no. 12, pp. 123 510–123 510–3, Mar. 2011.

[144] C. Augustine, A. Raychowdhury, D. Somasekhar, J. Tschanz, V. De, and K. Roy, "A Self-Consistent Simulation Framework for Spin-Torque Induced Domain Wall Propagation," *EDL (under review)*, 2011.

[145] X. Fong, S. K. Gupta, N. N. Mojumder, S. H. Choday, C. Augustine, and K. Roy, "KNACK: A hybrid spin-charge mixed-mode simulator for evaluating different genres of spin-transfer torque MRAM bit-cells," in *Simulation of Semiconductor Processes and Devices (SISPAD), 2011 International Conference on*, Sep. 2011, pp. 51 –54.

[146] N. Mojumder, C. Augustine, and K. Roy, "Self-Consistent Transport-Magnetic Simulation and Benchmarking of Hybrid Spin-Torque Driven Magnetic Tunnel Junctions (MTJs)," in *Proceedings of the Biennial University/Government/Industry Micro/Nano Symposium*, Jul. 2010, pp. 1 –6.

[147] C. Augustine, A. Raychowdhury, B. Behin-Aein, S. Srinivasan, J. Tschanz, V. De, and K. Roy, "Numerical analysis of domain wall propagation for dense memory arrays," in *Proceedings of the International Electron Devices Meeting*, Dec. 2011, pp. 17.6.1 –17.6.4.

VITA

VITA

Rangharajan Venkatesan received the B.Tech. degree in Electronics and Communication Engineering from the Indian Institute of Technology, Roorkee in 2009. Currently, he is pursuing the Ph.D. degree in Electrical and Computer Engineering at Purdue University. His research interests include spintronic memories, neural networks, approximate computing, and variation-tolerant design. He was a recipient of Purdue's Ross Fellowship for the year 2009-2010, the Bilsland Dissertation Fellowship for the year 2013-2014 and Intel's Spontaneous Recognition Award in 2013. His work on spintronic memory design was recognized with the Best Paper Award at the International Symposium on Low Power Electronics and Design (ISLPED), 2012 and the Best Presentation Award from The Center for Spintronic Materials, Interfaces and Novel Architectures (C-SPIN) in 2014. He has received the Best Undergraduate Thesis Award from Indian Institute of Technology, Roorkee in 2009 for his work exploring the impact of variations on FinFET-based SRAM.